NP IS EQUAL TO CO-NP*

Raju Renjit. G,

rajurenjitgrover@yahoo.com, Grover house, Tripunithura, Eranakulam, Kerala, India.

Abstract

In this endeavor, we first prove that all Turing machines that solves the clique problem are of a particular type using the method of backward induction. We then show that NP is equal to co-NP as an application of machine combinatorics.

*Not by might, nor by power, but by my SPIRIT, saith the LORD of hosts.

Contents

1 Introduction

A

- Clique
  - Is:
    - ▷ A complete
      - – Graph.
- The:

"clique problem"

can

- Be
  - Stated
    - ▷ As,
      - – Given:

"a finite graph,"

does

- It
  - Contain:

"a clique"

of

- Some
  - Particular:
    - ▷ Size?
- In

- This:
  - Endeavor,

we

- First
  - Prove
    - That,
      - All:

“*Turing machines*”

that

- Solves:
  - The
    - Clique
      - Problem

have

- A particular:

“*structure.*”

- And
  - Then
    - Using
      - That:

“*result*”

we

- Prove:

$$NP \quad = \quad co\text{-}NP.$$

- It
    - Is proved
        - In:
            - Ullman [1],

that,

- The
    - Clique
        - Problem
            - Is:

"*NP-Complete.*"

- It
    - Can
        - Be:
            - Shown

that,

- It
    - Is
        - Always:
            - Possible

to

- Check
    - Whether:

"*a graph*"

- Has:

“*a clique*”

of

- A particular
    - Size:
        - ▹ In

a

- Finite
    - Number
        - ▹ Of:
            - *Steps.*
- And
    - So
        - ▹ If:

“*something*”

- Solves:

“*the clique problem,*”

then

- It
    - Will
        - ▹ Always:

“*halt,*”

after

- Checking

- Whether
    - The
        - Given:

"*graph*,"

has

- A clique
    - In:
        - It.
- In
    - Section 2,
        - We:
            - Use

the

- Method
    - Of:

"*backward induction*"

to

- Show
    - That,
        - All:

"*Turing machines*,"

that

- Solves:
    - The

- ▷ Clique
  - – Problem

should

- Choose:

$$\text{“}\textit{subgraphs,}\text{”}$$

of

- The given
  - ○ Graph:
    - ▷ One
      - – By one.
- We
  - ○ Will
    - ▷ Soon
      - – Define:

$$n//2.$$

- If:

$$G$$

- Is:

$$\text{“}\textit{a graph,}\text{”}$$

then

- We
  - ○ Define:

$$p$$

to

- Be
  - The
    - Number
      - Of:

"*subgraphs*"

- With:

"$n//2$ *vertices*"

- In:

$$G.$$

- In:
  - Section 4,

we

- Will
  - Show:
    - That,

there

- *Exist*
  - A Case
    - In
      - Which:

"*a clique*"

- With:

$$\text{“}n//2 \quad \textit{vertices}\text{”}$$

is

- The:
  - p^{th} subgraph
    - ▹ Chosen,
- Thereby
  - Proving:

$$NP \quad = \quad co\text{-}NP.$$

- And
  - We
    - ▹ Do:
      - It,

as

- An
  - Application
    - ▹ Of:

“*machine combinatorics,*”

which

- We
  - Present
    - ▹ In:
      - Section 3.
- A Turing machine:
  - Consists

of

- A single
  - Dimensional
    - Infinite
      - Tape:

“*divided*”

- Into:

“*cells*,”

- And:

“*a head*,”

- And
  - A finite
    - Set
      - Of:

“*quintuples*,”

- A special
  - *State*
    - Called:

“*the initial state*,”

- And
  - A finite
    - Number
      - Of:

“*terminal states*.”

- Each
    - *Cell*
        - On:

"the infinite tape"

can

- Contain
    - At most
        - One:
            - *Alphabet.*
- When
    - A finite
        - String
            - Is:

"placed"

on

- The
    - Tape
        - Of:

"a machine,"

then

- Its head
    - Will
        - Initially:
            - Point

to

- The
    - Left
        - Most:
            - *Alphabet*

of

- The
    - Given:
        - String.
- Then
    - The:
        - Head

after

- Scanning
    - It,
        - Will:
            - Replace it

with

- Another:
    - *Alphabet*,
- Or
    - With
        - The same:
            - *Alphabet*.
- And

  - Then:
    - ▹ Move

to

- The
  - Second:
    - ▹ *Alphabet*,
- And
  - Go to
    - ▹ Another:
      - *State*
- Or remain
  - In
    - ▹ The same:
      - *State*.
- It
  - Will
    - ▹ Then:
      - Scan

the

- Second
  - *Alphabet*,
    - ▹ And:
      - Replace it

with

- Another:

    - *Alphabet*
- Or with
    - The
        - Same:
            - *Alphabet*,
- And
    - Then:
        - Move

to

- The:
    - Left
        - Or:
            - Right,
- And
    - Go to
        - Another:
            - *State*,
- Or remain
    - In
        - The same:
            - *State*.
- And
    - The process
        - Will:
            - Continue.
- These

    - Moves:
        - Made

is

- Defined
    - By
        - A finite:
            - Set

of

- Quintuples
    - Of:

“*the machine.*”

- This
    - Finite
        - Set
            - Of:

“*quintuples*”

is

- Called:

“*the Turing program*”

- Or:

“*the program*”

- Of:

“*the machine.*”

- And
  - So
    - If:

$$M$$

- Is:

"*a Turing machine*,"

- Then:

$$\chi(M)$$

is

- Defined
  - To
    - Be:

"*the program*"

- Of:

$$M.$$

- Exemplifying,
  - If:

$$\{\quad q_1,\quad q_2,\quad \ldots,\quad q_e\quad \}$$

is

- The
  - Set
    - Of
      - All:

“*quintuples*”

used

- By:

$$M$$

to

- Make:
    - *Moves*,
        - Then:

$$\chi(M) \quad = \quad \{ \quad q_1, \quad q_2, \quad \ldots, \quad q_e \quad \}.$$

- In
    - This:
        - Presentation,

it

- Is
    - *Not*:

“*required*”

to

- Know
    - How
        - A quintuple
            - Is:

“*represented.*”

- But

- It
  - Is
    - Only:

"*required*"

to

- Know
  - That
    - If:

$$M$$

- Is:

"*a Turing machine*,"

then

- Initially,
  - Its:
    - Head

will

- Point
  - To
    - The
      - Left most:

"*alphabet*"

of

- The
  - Given:

            - ▹ String,

- And
    - ◦ Also:
        - ▹ It

will

- Always
    - ◦ Start
        - ▹ From:

"*the initial state*"

- And
    - ◦ The
        - ▹ Quintuples
            - In:

$$\chi(M)$$

are

- Used
    - ◦ To
        - ▹ Make:
            - Moves

until

- It:

"*halts.*"

- See:
    - ◦ Ullman [1]

for

- More
  - Information
    - On:

$$\text{“\textit{Turing machines.}”}$$

- Also
  - The
    - Symbols:

$$\Rightarrow, \quad \in, \quad \subset, \quad \cup, \quad \setminus, \quad \emptyset \quad \textit{and} \quad \times$$

are

- Used
  - To denote:
    - Implication,
    - Set membership,
    - Is a subset of a set,
    - Set union,
    - Set difference,
    - Empty set,
    - Integer multiplication
      - Respectively.

2 Machine structure

Definition 1. $n//2$

- Is:
  - Defined

to

- Be:
  - $\frac{n}{2}$ if n is even,
  - $\frac{n-1}{2}$ if n is odd.
- And
  - If:

$$S$$

- Is:

“*a set,*”

- Then:

$$|S|$$

is

- Defined
  - To
    - ▷ Be:

“*the cardinality*”

- Of:

$$S.$$

Assumption 1. The

- Number
  - Of:
    - ▷ Vertices

in

- All
  - Graphs
    - Is:
      - Equal

to:

$$n,$$

- And:

$$G$$

is

- An
  - Arbitrary:

“*graph*”

- With:

“n *vertices.*”

Also:

$$V(G)$$

is

- The
  - Set
    - Of
      - All:

“*vertices*”

- In:

$$G,$$

- And:

$$G_a \subset G$$

- Means:

$$G_a$$

is

- A subgraph
  - Of:

$$G.$$

- We saw that:

"*a Turing machine*"

can

- Only
  - Work
    - With:
      - Strings.
- And
  - So:
    - It

cannot

- Straight
  - Away
    - Work

- On:

"a graph."

- And
  - So
    - For:

"a Turing machine"

to

- Work
  - On:

"a graph,"

there

- Should
  - Be
    - Some:
      - Means

to

- Convert:

"a graph"

- Into:

"a string."

- And
  - So
    - The

– Function:

$$E$$

is

- Used
  - To
    - Return:

“*a finite string*”

for

- A given
  - Graph:

$$G.$$

- But
  - At
    - This:
      - Point,

we

- *Cannot*
  - Assume:
    - That,

the

- String
  - Returned:
    - By:

$$E$$

is

- A representation
    - Of
        - ▹ The
            - Given:

“*graph*.”

- We
    - Can
        - ▹ Only:
            - Assume

that,

- The
    - String
        - ▹ Returned:
            - By:

$$E$$

is

- Some
    - String
        - ▹ Constructed:
            - In some way.
- And
    - So:

$$E$$

will

- Simply
    - Return:

"a string"

for

- A given:

"graph"

on

- Which
    - A Turing machine
        - Can:
            - Work.
- And so
    - The string
        - Returned:
            - By:

$$E$$

may

- Or
    - May
        - *Not*
            - Be:

"a representation"

of

- The
    - Given:

"graph."

- We
    - Believe:
        - That,

if

- We
    - Assume:
        - That,

the

- String
    - Returned:
        - By:

$$E$$

is

- Indeed:

"a representation"

of

- The
    - Given:

"graph,"

- And
  - The length
    - Of
      - That:

"*string*"

is

- A polynomial
  - Of
    - The number
      - Of:

"*vertices*,"

then

- It
  - Will
    - Complicate
      - The:

"*proof*."

- And
  - So:
    - When

we

- Attempt
  - To:
    - Prove

that,

- All:

"*Turing machines*"

that

- Solves:

"*the clique problem*"

have

- A particular:

"*structure,*"

the

- Question:

"*is the length of the string*

returned by E

a polynomial of

the number of vertices in the graph?"

should

- *Not*
  - Be
    - ▹ Taken
      - *Into*:

"*consideration.*"

Definition 2. The

- Function:

$$E$$

is

- Defined
  - To
    - Be:

“*a computable function,*”

such that

- It
  - Takes:

$$G$$

as

- The
  - Only:
    - Parameter,
- And
  - Returns:

“*a string,*”

- And
  - For
    - *Two*
      - Graphs:

$$G_1 \quad \textit{and} \quad G_2,$$

- If:

$$G_1 \qquad \textit{and} \qquad G_2$$

have

- The
    - Same:

$$\text{"\textit{adjacency matrix},"}$$

- Then:

$$E(G_1) \quad = \quad E(G_2),$$

- Else:

$$E(G_1) \quad \neq \quad E(G_2).$$

So we see that,

- The
    - Value
        - ▷ Returned:
            - By:

$$E$$

for

- *Two*
    - *Graphs*
        - ▷ Will
            - Be:

$$\text{"\textit{the same},"}$$

if

- And
    - Only
        - ▷ If
            - Those:

“*two graphs*”

have

- The
    - Same:

“*adjacency matrix.*”

- And
    - So:

$$E$$

- Is:

“*a bijective function.*”

- And
    - So:
        - ▷ Initially,

we

- Pass
    - The
        - ▷ Given:

“*graph*”

- To:

$$E$$

so

- As
  - To
    - ▹ Get
      - The string:

S.

- And
  - Then
    - ▹ Place:
      - It

on

- The
  - Tape
    - ▹ Of:

"*a Turing machine*"

that

- Solves:

"*the clique problem*,"

- And
  - If
    - ▹ That
      - Machine:

"*accepts*"

- The:

$$\text{“}\textit{string,}\text{”}$$

then

- That
  - Graph
    - Will:
      - Have

a

- Clique
  - Of
    - Size:

$$n//2,$$

- Else
  - That:
    - Graph

will

- *Not*
  - Have
    - Such:

$$\text{“}\textit{a clique.}\text{”}$$

Note that,

- The
  - Set
    - Of

- All:

$$E$$

may

- Or
  - May
    - *Not*
      - Be:

"*infinite*."

- But
  - The
    - Same:

$$E$$

will

- Always
  - Used
    - To:
      - *Construct*

the

- String
  - Given
    - To:

"*the Turing machine*"

that

- Solves:

“*the clique problem*.”

Definition 3. Let:

$$M$$

- Be:

“*a Turing machine*,”

- And
  - Let:

S

- Be:

“*a string*.”

- If:

$$M$$

- Accepts:

S

- And:

“*halts*,”

- Then:

$$M(\text{S}) \quad = \quad 1.$$

- If:

$$M$$

- Rejects:

$$\mathbf{S}$$

- And:

$$\textit{“halts,”}$$

- Then:

$$M(\mathbf{S}) \quad = \quad 0.$$

So

- We
  - Are:
    - Going

to

- Give
  - A precise
    - Definition
      - For:

$$\textit{“a Turing machine”}$$

that

- Solves:

$$\textit{“the clique problem.”}$$

Definition 4. Let:

$$E(G) \quad = \quad \mathbf{S}.$$

- Then:

$$M$$

is

- Defined
  - To
    - Be:

"*a Turing machine,*"

- Such that:

$$M(\mathbf{S}) \quad = \quad 1,$$

if

- There
  - Is:

"*a clique*"

of

- Size:

$$n//2$$

- In:

$$G,$$

- And:

$$M(\mathbf{S}) \quad = \quad 0,$$

if

- There
  - Is
    - *No*
      - Such:

$$\text{“}\textit{clique}\text{”}$$

- In:

$$G,$$

- And:

$$\text{“}M \quad \textit{halts}\text{”}$$

for

- All:

$$\text{“}\textit{strings.}\text{”}$$

So we see that,

- Intuitively:

$$M$$

is

- The
    - Place
        - Where
            - The:

$$\text{“}\textit{computation}\text{”}$$

is

- Done
    - To
        - Say
            - Whether:

$$G$$

- Contains:

"*a clique*"

of

- Size:

$$n//2.$$

- And so
  - From
    - ▹ Now:
      - Onwards,

we

- Assume
  - That:

$$E$$

is

- An
  - Arbitrary:

"*function*"

as

- Defined
  - Above,
- And:

$$M$$

an

- Arbitrary:

"*Turing machine*"

as

- Defined:
  - Above.

Assumption 2. Unless

- Otherwise
  - Stated:

$$G_c$$

- Is:

"*a clique*"

of

- Size:

$$n//2$$

- In:

$$G.$$

- And:

S

- Is:

"*a string,*"

- Such that:

$$E(G) \quad = \quad \text{S}.$$

So

- From
    - The
        - ▹ Above:
            - Assumption 2,

we see that,

$$M(\text{S}) \quad = \quad 1.$$

- And
    - So
        - ▹ Initially,
            - When:

$$\text{S}$$

is

- Placed
    - On
        - ▹ The:

"*tape*"

- Of:

$$M,$$

its

- Head
    - Will:
        - ▹ Point

to

- The
  - Left most
    - *Alphabet*
      - Of:

S.

- Then
  - At
    - This:
      - Point,

we

- Say
  - That:

$$M$$

is

- In
  - The
    - Initial:
      - Configuration.
- Then
  - The
    - Head
      - Of:

$$M$$

will

- Read
  - That:
    - *Alphabet*,
- And
  - Replace:
    - It,
- And
  - Move
    - To:
      - The right.
- Then
  - We
    - Say
      - That:

$$M$$

has

- *Entered*
  - Into:

“*the second configuration*.”

- And
  - Also
    - At:
      - This time,

the

- Head
  - Will
    - Be:
      - Scanning

the

- Second:
  - *Alphabet*
    - Of:

$$S.$$

- Then:

$$M$$

will

- Replace
  - That:
    - *Alphabet*,
- And
  - Move
    - One:
      - Step

to

- The:
  - Left
    - Or:

- Right,

- And
  - Enter
    - The third:
      - Configuration.
- And
  - So
    - Let:

$$\alpha_1, \quad \alpha_2 \quad \textit{and} \quad \alpha_3$$

be

- Used
  - To:
    - Denote

the

- Initial,
  - Second,
    - And third:
      - Configurations.
- Then
  - Since:

$$\alpha_3$$

occurs

- After:

$$\alpha_2$$

which

- Inturn
    - Occurs
        - After:

$$\alpha_1$$

we

- Write:

$$\alpha_1 \quad \rightarrow \quad \alpha_2 \qquad \textit{and} \qquad \alpha_2 \quad \rightarrow \quad \alpha_3.$$

- And
    - We
        - Write:

$$\alpha_1 \quad \rightarrow \quad \alpha_2 \quad \rightarrow \quad \alpha_3$$

as

- An
    - Abbreviation
        - For:

$$\alpha_1 \quad \rightarrow \quad \alpha_2 \qquad \textit{and} \qquad \alpha_2 \quad \rightarrow \quad \alpha_3.$$

- And
    - So
        - In:
            - General,

if

- The

- ◦ Configuration:

$$\alpha_{i+1} \quad \textit{follows after} \quad \alpha_i,$$

we

- Write:

$$\alpha_i \quad \rightarrow \quad \alpha_{i+1}.$$

Definition 5. Let:

S

- Be:

“*placed*”

on

- The
  - Tape
    - Of:

$$M,$$

- And
  - Let:

$$M$$

start

- From
  - The
    - Configuration:

$$\alpha_1,$$

- And
  - Let:

$$\alpha_1 \to \alpha_2 \to \alpha_3 \to \cdots \to \alpha_i \to \alpha_{i+1} \to \cdots \to \alpha_k \to \alpha_{k+1},$$

- And:

$$\text{“}M \quad \textit{halts}\text{”}$$

on

- Reaching:

$$\alpha_{k+1}.$$

Then:

$$C_{\text{S}}$$

is

- Defined
  - To
    - Be
      - The sequence:

$$\alpha_1, \quad \alpha_2, \quad \alpha_3, \quad \ldots, \quad \alpha_i, \quad \alpha_{i+1}, \quad \ldots, \quad \alpha_k, \quad \alpha_{k+1}.$$

But we see that,

- If
  - We
    - Are
      - To:

“*write*”

a

- Proof
    - Using
        - The:
            - List

of:

"*configurations*,"

then

- It
    - Will
        - Complicate:
            - The

"*proof*,"

since

- Some
    - Parts:
        - Of

the

- Given
    - String
        - May:
            - *Not*

be

- Analyzed
    - By:

$$M.$$

- And
    - So
        - All
            - Those:

$$\text{"}\textit{unanalyzed parts}\text{"}$$

of

- The
    - String
        - Should:
            - Also

have

- To
    - Be:
        - Considered.
- But
    - We see that,
        - In:

$$C_{\text{S}},$$

the

- Move
    - From:

$$\alpha_i \quad \textit{to} \quad \alpha_{i+1}$$

is

- Always
    - Made:
        - ▹ Using

a

- Particular
    - Element
        - ▹ Of:

$$\chi(M).$$

- And
    - Similarly,
        - ▹ A particular:
            - Quintuple

will

- Be
    - Used to
        - ▹ Move
            - From:

$$\alpha_{i+1} \qquad \textit{to} \qquad \alpha_{i+2}.$$

- And
    - So
        - ▹ For
            - All:

$$C_{\mathrm{S}},$$

there

- Will
    - Be
        - ▹ A unique:
            - – *Sequence*

of

- Quintuples
    - To
        - ▹ Which:

$$C_{\mathrm{S}}$$

can

- Be:

“*mapped.*”

- And
    - That
        - ▹ Sequence
            - – Will:

“*contain*”

all

- The:

“*details*”

of

- The
    - Computation:
        - ▹ Done.

- And
    - So
        - Let:

$$q_i \;\in\; \chi(M)$$

be

- Used
    - To:
        - Make

the

- Move
    - From:

$$\alpha_i \quad to \quad \alpha_{i+1}.$$

- Then
    - We
        - Write:

$$\alpha_i \xrightarrow{q_i} \alpha_{i+1}$$

- And
    - So
        - With:
            - This

we

- Construct:

$$Q\text{s}$$

which

- Is
    - The
        - *Exact*:
            - Sequence

of

- Quintuples
    - Used
        - To
            - Construct:

$$C_{\mathrm{S}}.$$

Definition 6. Let:

$$C_{\mathrm{S}}$$

- Be:

$$\alpha_1, \quad \alpha_2, \quad \alpha_3, \quad \ldots, \quad \alpha_i, \quad \alpha_{i+1}, \quad \ldots, \quad \alpha_k, \quad \alpha_{k+1}.$$

- And
    - Let:

$$\alpha_1 \quad \xrightarrow{q_1} \quad \alpha_2 \quad \xrightarrow{q_2} \quad \alpha_3 \quad \ldots \quad \alpha_i \quad \xrightarrow{q_i} \quad \alpha_{i+1} \quad \ldots \quad \alpha_k \quad \xrightarrow{q_k} \quad \alpha_{k+1}.$$

- Then:

$$Q_{\mathrm{S}}$$

is

- Defined
    - To

- ▹ Be:

$$q_1, \quad q_2, \quad \ldots, \quad q_i, \quad \ldots, \quad q_k.$$

Note that,

- There:
  - ◦ May
    - ▹ Or
      - May *not*

be

- Repetitions
  - ◦ In:

$$Q_{\mathrm{S}}.$$

- Also
  - ◦ Since:

$$Q_{\mathrm{S}}$$

is

- A finite
  - ◦ Sequence
    - ▹ Of:
      - Quintuples,

we see that,

- It
  - ◦ Has:

“*a length.*”

- And
    - So:

$$|Q_{\mathrm{S}}|$$

is

- Defined:
    - To
        - Be
            - The:

"*length*"

- Of:

$$Q_{\mathrm{S}}.$$

Definition 7. Let:

$$Q_{\mathrm{S}}$$

be

- The
    - Sequence:

$$q_1, \quad q_2, \quad q_3, \quad \ldots, \quad q_{k-1}, \quad q_k, \quad q_{k+1}, \quad \ldots, \quad q_{m-1}, \quad q_m.$$

- Then:

$$k\text{-}Q_{\mathrm{S}}$$

is

- Defined
    - To
        - Be:

$$q_1, \quad q_2, \quad q_3, \quad \dots, \quad q_{k-1}, \quad q_k.$$

Also:

$$q^*$$

is

- Defined
    - As
        - ▷ Some
            - Finite:

“*sequence*”

of

- Elements
    - Of:

$$\chi(M).$$

- And
    - So:

$$q^*$$

maybe

- Of
    - Length:
        - ▷ *Zero.*
- And
    - Also:
        - ▷ There

may

- Or
  - May:
    - *Not*

be

- Repetitions
  - In:

$$q^*.$$

- This
  - Presentation:
    - Deals

with

- The
  - Computational
    - Complexities:
      - Related

to

- The
  - Question:

“does a graph

contains a clique

of size $n//2$?”

- And

- So
  - We
    - Define:

$$\Phi$$

- As:

“*a function*”

which

- Checks
  - Whether
    - A given:
      - Graph

is:

“*a clique.*”

- And
  - So
    - If:

$$\Phi(G_a) \quad = \quad true,$$

- Then:

$$G_a$$

- Is:

“*a clique.*”

- But
  - If:

- ▷ *Not*,
  - Then:

$$G_a$$

- Is:

"*not a clique*."

- Also
  - For
    - ▷ The:
      - Sake

of:

"*simplicity*,"

we say that,

- *No*
  - Function
    - ▷ Other
      - Than:

$$\Phi$$

can

- Produce
  - The
    - ▷ Same:
      - Result

of:

$$\Phi.$$

- Also
  - When
    - We
      - Say:

“*something*”

need

- *Not*
  - Be:

“*done*,”

we mean that,

- That:

“*something*”

may

- Or
  - May
    - *Not*
      - Be:

“*done*.”

- But if
  - It
    - Will
      - Be:

“*done*,”

then

- It
  - Will
    - *Not*
      - Be:

"*used.*"

Assumption 3. If:

"*something*"

need

- *Not*
  - Be:

"*done,*"

then

- It
  - Will
    - *Not*
      - Be:

"*done.*"

2.1 Sketch of proof

- Since $M(\mathrm{S}) = 1$, we see that, Q_{S} must imply $M(\mathrm{S}) = 1$.
- Also since M halts as soon as it finds that $G_c \subset G$, we see that, there is a minimal last part of Q_{S} which implies $M(\mathrm{S}) = 1$.
- Let this minimal last part of Q_{S} which implies "$M(\mathrm{S}) = 1$" be called ${}^0Q_{\mathrm{S}}$.
- We then proved that, G_c can be deduced from ${}^0Q_{\mathrm{S}}$.

- The intuitive meaning of ${}^{0}Q_{\mathrm{S}}$ is that, it is the minimal last part of Q_{S} from which G_c and 0 subgraphs of G that are *not* cliques of size $n//2$ can be deduced.

- We then proved that, if there is a quintuple in Q_{S} that is *not* a part of ${}^{0}Q_{\mathrm{S}}$, then it is possible to deduce at least 1 subgraph of G which is *not* a clique of size $n//2$ from Q_{S}.

- So we define ${}^{1}Q_{\mathrm{S}}$ (conditionally) as the minimal last part of Q_{S} from which G_c and 1 subgraph of G which is *not* a clique of size $n//2$ can be deduced.

- Similarly, it can be shown that, if there is a quintuple that is *not* a part of ${}^{1}Q_{\mathrm{S}}$ in Q_{S}, then at least 2 subgraphs which are *not* cliques of size $n//2$ can be deduced from it.

- So we define ${}^{2}Q_{\mathrm{S}}$ (conditionally).

- And similarly, we define ${}^{3}Q_{\mathrm{S}}$, ${}^{4}Q_{\mathrm{S}}$, $\ldots$, ${}^{j}Q_{\mathrm{S}}$ (conditionally).

- So now we know that ${}^{0}Q_{\mathrm{S}}$ certainly exists, but we are *not* sure about the existence of: ${}^{1}Q_{\mathrm{S}}$, ${}^{2}Q_{\mathrm{S}}$, ${}^{3}Q_{\mathrm{S}}$, $\ldots$, ${}^{j}Q_{\mathrm{S}}$.

- So we know that j is at least 0, since we know that ${}^{0}Q_{\mathrm{S}}$ exists.

- But we do not know what the maximum value j can have.

- And so if j is never greater that 0, then in all cases, when S is placed on the tape of M, it will immediately choose G_c with out choosing any other subgraph of G.

- But if in some cases, j does have a positive *non-zero* value, we can say that, M will choose (at least one or) some subgraphs of G before choosing G_c.

- So we show that, in some cases, there is a quintuple in Q_{S} that is *not* a part of ${}^{0}Q_{\mathrm{S}}$.

- Thereby proving that ${}^{1}Q_{\mathrm{S}}$ exists in at least one case.

- We then iterate this step to prove that ${}^{j}Q_{\mathrm{S}}$ exists (for some unknown value of j).

- And so from this, we see that, in some cases, M will choose subgraphs of G one by one, until G_c has been chosen; thereby proving that M indeed has a structure.

2.2 Proving the structure

Let:

$$f$$

- Be:

"*a deterministic function*."

- Then
  - If
    - ▹ It
      - Performs:

"*a computation*"

without

- Using
  - Its:
    - ▹ Parameters,

then

- That:

"*computation*"

will

- Be
  - A constant:
    - ▹ Computation.

- Or:

$$f$$

will

- Always
    - Start
        - From:
            - One point,
- And will
    - Always
        - End up:
            - In another.
- And
    - So
        - In
            - Such cases:

$$f$$

cannot

- Say
    - Anything
        - About
            - Its:

"*parameters.*"

- But if
    - It
        - Uses

- Its:

"parameters"

to

- Perform:

"a computation,"

then

- It
  - Will:
    - Look

at

- Something
  - Of
    - Its:
      - Parameter,
- And
  - Make
    - A move:
      - Accordingly.
- And
  - So
    - From:
      - This,

we see that,

- If

- The
  - ▹ Value:
    - – Returned

by:

"*a function*"

has

- To
  - Imply
    - ▹ Something:
      - – About

its:

"*parameters,*"

then

- That
  - It:
    - ▹ Should

do

- Some
  - Operation
    - ▹ On
      - – Its:

"*parameters.*"

Statement 1. *All*

- *Non-constant*
  - *Functions*

            ▷ *Will*
                - *Have:*

"a parameter,"

- *And*
    ○ *All*
        ▷ *Values:*
            - *Returned*

by:

"a non-constant function"

will

- *Depend*
    ○ *On*
        ▷ *The*
            - *Given:*

"parameters."

Statement 2. *If*

- *Some*
    ○ *Computation*
        ▷ *Done*
            - *By:*

$$M$$

has

- *To*
    ○ *Imply*

▹ *Something*

- *About:*

$$G,$$

then

- *That*
  - *Computation:*
    - ▹ *Must*

be

- *Done*
  - *On:*

$$G.$$

Let:

$$M, \quad E \quad \text{and} \quad G$$

- Remain:

“*fixed*,”

- And
  - Assume:

$$A \quad \Rightarrow \quad B.$$

- Then
  - We see that,
    - ▹ There
      - Is:

“*something*”

- In:

$$A$$

that

- Can
  - Be

“*transformed*”

- To:

$$B,$$

using

- The
  - Axioms
    - Of:
      - Logic,
- And
  - Of
    - The:
      - System.
- And
  - So
    - That:

“*something*”

- Of:

$$A$$

can

- Be
  - Considered:
    - As

a

- Representation
  - Of:

$$B.$$

- And
  - So
    - Let:

$$k\text{-}Q_{\mathrm{S}}$$

- Be:

“*defined.*”

- And
  - Assume:

$$k\text{-}Q_{\mathrm{S}} \quad \Rightarrow \quad G_a \subset G \quad \textit{and} \quad \Phi(G_a) \quad = \quad true \text{ or } false. \qquad (1)$$

- Then since
  - We
    - Assumed:
      - Equation 1,

we see that,

- There

- Will:
  - *Exist*

a

- Minimal
  - Sub sequence
    - In:

$$k\text{-}Q\text{s},$$

- Say:

$$Q_{G_a}$$

- Such that:

$$Q_{G_a} \quad \Rightarrow \quad G_a \subset G \quad \textit{and} \quad \Phi(G_a) \; = \; true \text{ or } false.$$

- Then
  - Since:

$$G_a \subset G \quad \textit{and} \quad \Phi(G_a) \; = \; true \text{ or } \; false$$

can

- Be
  - Deduced
    - From:

$$Q_{G_a},$$

- We see that:

$$Q_{G_a}$$

can

- Be
  - Considered:
    - As
      - Some:

"*representation*"

- Of:

$$G_a \subset G \quad \textit{and} \quad \Phi(G_a) \;=\; true \text{ or } false.$$

- Then
  - Since:

$$M, \quad E \quad \textit{and} \quad G$$

- Are:

"*fixed,*"

we see that,

- The
  - Sub sequence:

$$Q_{G_a}$$

will

- Be
  - Unique
    - For:

$$G_a,$$

- And
  - So:

▷ We

can

- Consider:

$$Q_{G_a}$$

as

- A representation
  - Of:

$$G_a.$$

- And
  - So
    - ▷ We
      - Can:

$$\text{“}\textit{construct}\text{”}$$

a

- Function,
  - Say:

$$f$$

such that

- It
  - Will
    - ▷ Check
      - Whether:

$$Q_{G_a}$$

is

- A sub sequence
    - Of:

$$k\text{-}Q\text{s}.$$

- Or
    - Since:

$$Q_{G_a}$$

- Is:

"*unique*,"

- We see that:

$$Q_{G_a}$$

can

- Be
    - Inbuilt
        - Into:

$$f.$$

- And so
    - It
        - Can
            - Simply:

"*check*"

- Whether:

$$k\text{-}Q_\mathrm{S}$$

has

- That
    - Sub sequence
        - ▹ In:
            - It.
- And
    - So:
        - ▹ There

will

- *Exist*:

“*a function*”

- Say:

$$f,$$

such that

- It
    - Will
        - ▹ Take:

$$k\text{-}Q_\mathrm{S},$$

- And
    - Checks
        - ▹ Whether:

$$Q_{G_a}$$

is

- A sub sequence
    - Of:

$$k\text{-}Q\text{s},$$

- And
    - If
        - It:
            - Finds

that:

$$Q_{G_a}$$

is

- Indeed:
    - A sub sequence
        - Of:

$$k\text{-}Q\text{s},$$

then

- It
    - Will
        - Return:
            - A set

with:

$$Q_{G_a}$$

as

- The

- Only
  - Element:
    - In it,

- But
  - If:
    - *Not*,

it

- Will
  - Return:

"*a null set*."

- And
  - So
    - When:
      - Equation 1

is:

"*true*,"

then

- There
  - Will:
    - *Exist*

a

- Computable:

"*function*,"

such that

- It
  - Will
    - ▷ Take:

$$k\text{-}Q_{\mathrm{S}}$$

as

- The
  - Only:
    - ▷ Parameter,
- And
  - Return
    - ▷ A set
      - – With:

$$G_a$$

as

- The
  - Only
    - ▷ Element:
      - – In it.

Lemma 1. *When*

- *The*
  - *Statement:*

$$k\text{-}Q_S \quad \Rightarrow \quad G_a \subset G \quad \textit{and} \quad \Phi(G_a) \;=\; true\ \textit{or}\ false$$

holds,

- *There*

- *Will:*
    - *Exist*

a

- *Computable*
    - *Function,*
        - *Say:*

$$f$$

- *Such that:*

$$f(k\text{-}Q_S) \quad = \quad \{ \quad G_a \quad \}.$$

Then

- Since
    - Assumption 2
        - Say
            - That:

$$G_c \subset G,$$

- We see that:

$$M(\mathbf{S}) \quad = \quad 1.$$

- But
    - We see that:

$$M(\mathbf{S}) \quad = \quad 1$$

implies

- The
    - Existence

▷ Of:

$$Q_{\text{S}}.$$

- And
  - Also:

$$Q_{\text{S}}$$

- Implies:

$$M(\text{S}) \quad = \quad 1.$$

- And
  - So:

$$(M(\text{S}) \quad = \quad 1) \quad \Rightarrow \quad Q_{\text{S}} \quad \textit{and} \quad \textit{vice versa.} \tag{2}$$

- And
  - Also:

$$(M(\text{S}) \quad = \quad 1) \quad \Rightarrow \quad G_c \subset G \ \ \textit{and} \ \ \Phi(G_c) \quad = \quad true. \tag{3}$$

- And
  - So
    - ▷ From:
      - – Equations 2 and 3,

we see that:

$$Q_{\text{S}} \quad \Rightarrow \quad G_c \subset G \quad \textit{and} \quad \Phi(G_c) \quad = \quad true.$$

- And
  - So
    - ▷ We see that:

$$k\text{-}Q_{\text{S}} \quad \Rightarrow \quad G_c \subset G \quad \textit{and} \quad \Phi(G_c) \quad = \quad true,$$

- Where:

$$k$$

is

- The
  - Length
    - Of:

$$Q_{\mathrm{s}}.$$

- And
  - So
    - From:
      - These,

- And
  - From:
    - Lemma 1,

we see that,

- There
  - Will:
    - *Exist*

a

- Computable
  - Function,
    - Say:

$$f$$

- Such that:

$$f(Q_{\mathrm{S}}) \quad = \quad \{ \quad G_c \quad \}.$$

Lemma 2. *There*

- *Exists:*
  - *A computable:*
    - *Function,*
      - *Say:*

$$f$$

- *Such that:*

$$f(Q_S) \quad = \quad \{ \quad G_c \quad \}.$$

So we see that,

- When:

$$M$$

- Accepts:

“*a string*,”

then

- There
  - Will:
    - Be

at

- Least
  - *One*
    - Sub sequence
      - Of:

$$Q\text{s}$$

which

- Implies:

$$G_a \subset G \quad \textit{and} \quad \Phi(G_a) \;=\; true \text{ or } false.$$

- And
    - So
        - At:
            - Present,

we

- Know
    - That
        - There:
            - Is

at

- Least
    - *One*
        - Such:

"*sub sequence.*"

- Or
    - At:
        - Present,

we

- Know
    - That:

$$\text{“}Q_{G_c} \quad \textit{exists.”}$$

- And
  - So
    - ▹ We:
      - Can

write:

$$Q_\text{S}$$

- As:

$$q^* \quad Q_{G_c} \quad q^*.$$

- But
  - We see that,
    - ▹ There

may

- Or
  - May:
    - ▹ *Not*

be

- Other
  - Sub sequences
    - ▹ In:

$$Q_\text{S}$$

which

- Implies

- Such:
  - *Statements.*
- And
  - So
    - If:

$$S_{\text{S}}$$

is

- The
  - Set
    - Of:
      - All

such

- Minimal:

“*sub sequences,*”

- And
  - If:

$$S_{\text{S}} \;=\; \{\; Q_{G_a},\;\; Q_{G_b},\;\; Q_{G_c} \;\},$$

- Then:

$$Q_{\text{S}}$$

can

- Be
  - Written
    - As
      - Either:

$$q^* \quad Q_{G_a} \quad q^* \quad Q_{G_b} \quad q^* \quad Q_{G_c} \quad q^*$$

or

$$q^* \quad Q_{G_b} \quad q^* \quad Q_{G_a} \quad q^* \quad Q_{G_c} \quad q^*.$$

- And so
  - Let
    - ▹ Us
      - – Define:

$$I_{\text{S}}.$$

- We
  - Define:

$$I_{\text{S}}$$

as

- The
  - *Exact*
    - ▹ Sequence
      - – Of:

"*quintuples*"

- In:

$$Q_{\text{S}}$$

which

- Occurs
  - Before
    - ▹ Any:

- *Element*

of:

$$S_{\text{S}}.$$

- And
  - So
    - If:

$$S_{\text{S}} \quad = \quad \{ \quad Q_{G_a}, \quad Q_{G_b}, \quad Q_{G_c} \quad \},$$

- Then:

$$Q_{\text{S}}$$

can

- Be
  - Written
    - As
      - Either:

$$I_{\text{S}} \quad Q_{G_a} \quad q^* \quad Q_{G_b} \quad q^* \quad Q_{G_c} \quad q^*$$

or

$$I_{\text{S}} \quad Q_{G_b} \quad q^* \quad Q_{G_a} \quad q^* \quad Q_{G_c} \quad q^*.$$

- But
  - It:
    - Maybe

the

- Case
  - That:

$$Q_{\text{S}}$$

may

- Start
  - With:

$$\text{“}\textit{an element}\text{”}$$

- Of:

$$S_{\text{S}}.$$

- Then
  - In
    - That:
      - Case,

we see that:

$$I_{\text{S}}$$

will

- Be
  - Of
    - Length:
      - *Zero.*
- And
  - So
    - Intuitively:

$$I_{\text{S}}$$

is

- The
  - Computation
    - ▹ Performed
      - Before:

$$M$$

- Chooses:

"*a subgraph.*"

- And
  - So
    - ▹ We:
      - Can

say

- That:

"Q_{S} *starts with* I_{S}

followed by

the of elements of S_{S}

and a single q^*."

- We
  - Saw:
    - ▹ That,

the

- Aim
  - Of:

$$M$$

is

- To
    - Check
        - ▹ Whether:

$$G$$

has

- A clique
    - Of
        - ▹ Size: $n//2$
            - In it,
- And
    - From:
        - ▹ Lemma 2,

we

- Know
    - That:

$$M$$

will

- Choose:

$$G_c$$

at

- Some

- ◦ Point
  - ▹ Of:
    - – Time.

- And
  - ◦ So
    - ▹ We:
      - – Can

assume

- That,
  - ◦ When:

$$M$$

finds

- That:

$$G_c$$

- Is:

"*a subgraph*"

- Of:

$$G,$$

then

- It
  - ◦ Will:
    - ▹ Take

the

- Necessary
    - Steps
        - To:

“*halt,*”

- And
    - Say:
        - That,

the

- String
    - Is:

“*accepted.*”

- And
    - So:
        - We

can

- Give:
    - A name

for

- The
    - Minimal:
        - Last
            - Part

of:

$$Q\text{s}$$

from

- Which:

$$G_c$$

can

- Be:

“*deduced.*”

- And
  - So
    - ▹ We
      - Define:

$${}^0Q\text{s}$$

as:

- The
  - Minimal:
    - ▹ Last
      - Part

of:

$$Q\text{s}$$

from

- Which:

$$G_c$$

- And: 0

- ◦ Subgraphs
  - ▹ Of:

$$G$$

that

- Does
  - ◦ *Not*
    - ▹ Satisfy:

$$\Phi$$

can

- Be:

“*deduced.*”

- Exemplifying,
  - ◦ If:

$$S_{\text{S}} \quad = \quad \{ \quad Q_{G_a}, \quad Q_{G_c} \quad \},$$

- Then:

$$Q_{\text{S}}$$

can

- Be
  - ◦ Written
    - ▹ As:

$$I_{\text{S}} \quad Q_{G_a} \quad q^* \quad {}^0Q_{\text{S}}.$$

- We

- Give a:

$$`0'$$

in

- The
  - Definition
    - Of:

$$^{0}Q_{\mathrm{S}}$$

since

- We
  - Will
    - Later
      - Define:

$$“^{1}Q_{\mathrm{S}} \quad (\textit{conditionally})”$$

$$“\textit{and so on.}”$$

Note that,

- We are
  - Just
    - Explaining
      - Things now,
- And
  - We
    - Do *not*
      - Define:

$$^{1}Q_{\mathrm{S}}$$

until

- We
  - Prove
    - That:

$$^{1}Q_{\mathrm{S}}$$

can

- Exist:
  - *Conditionally*.
- And
  - So
    - Informally:

$$^{1}Q_{\mathrm{S}}$$

is

- The
  - Minimal:
    - Last
      - Part

of:

$$Q_{\mathrm{S}}$$

from

- Which:

$$G_c$$

- And: 1

- Subgraphs
  - Of:

$$G$$

that

- Does
  - *Not*
    - Satisfy:

$$\Phi$$

can

- Be:

"*deduced.*"

- Exemplifying,
  - If:

$$S_{\text{S}} \quad = \quad \{ \quad Q_{G_a}, \quad Q_{G_b}, \quad Q_{G_c} \quad \},$$

- Then:

$$Q_{\text{S}}$$

can

- Be
  - Written
    - As
      - Either:

$$I_{\text{S}} \quad Q_{G_a} \quad q^{*} \quad {}^{1}Q_{\text{S}}$$

or

$$I_{\text{S}} \quad Q_{G_b} \quad q^* \quad {}^1Q_{\text{S}}.$$

- Then
  - Since
    - The:
      - Aim

of:

$$M$$

is

- To
  - Check
    - Whether:

$$G$$

has

- A clique
  - Of
    - Size: $n//2$
      - In it,
- And
  - Since:

$${}^0Q_{\text{S}}$$

- Is:

“*generated*”

if

- And
    - Only
        - If:

$$G_c$$

is

- A subgraph
    - Of:

$$G,$$

we see that,

- The
    - Ultimate
        - Aim
            - Of:

$$M$$

is

- To
    - Generate:

$$^0Q\text{s}.$$

- But
    - We see that,
        - Since:

$$M$$

- Is:

“*deterministic*,”

it

- Has
  - Some
    - Precise:
      - *Rules*

to

- Perform
  - All:
    - Computations.
- Or
  - We see that,
    - If:

$$M$$

does

- *Not*
  - Have:

“*a reason*”

- To
  - Perform:
    - A computation,

then

- It

- Will *not*
    - Perform:
        - It.

- And
    - So:

$$M$$

will

- *Not*
    - Generate:

$$^0Q_{\mathrm{S}},$$

unless

- And
    - Until
        - It:
            - Has

a

- Reason
    - To
        - Generate:

$$^0Q_{\mathrm{S}}.$$

- And
    - So
        - All:
            - Computations

done

- Before:

$$^{0}Q\text{s}$$

must

- Imply:

"*something*,"

- So that:

$$M$$

will

- Have
  - A reason
    - To
      - Generate:

$$^{0}Q\text{s}.$$

- And
  - So:
    - The sum
      - Total

of

- All
  - The
    - Things:
      - Implied

by

- All:

$$\text{“}\textit{computations}\text{”}$$

done

- Before:

$${}^{0}Q\text{s}$$

must

- Finally
  - Give
    - A reason
      - For:

$$M$$

to

- Generate:

$${}^{0}Q\text{s}.$$

- And
  - So:
    - The sum
      - Total

of

- All
  - The
    - Things:
      - Implied

by

- All
  - Computations
    - Done
      - Before:

$$^{0}Q\text{s}$$

must

- Imply:

“*something*”

- About:

$$G,$$

- So that:

$$M$$

will

- Have:
  - A reason
    - To:
      - Perform

a

- Computation
  - On
    - The:
      - Part

of:

$$G$$

- Containing:

$$G_c.$$

- And
    - So
        - From:
            - This,
- And
    - From:
        - Statement 2,

we see that,

- All
    - Computations
        - Done
            - By:

$$M$$

will

- Be
    - Only
        - On:

$$G.$$

Statement 3. *All*

- *Computations*
    - *Done*
        - *By:*

$$M$$

will

- *Be*
  - *Only*
    - *On:*

$$G.$$

We

- Saw
  - That:

$$I_{\mathrm{S}}$$

is

- Done
  - Before:

$$M$$

- Chooses:

“*a subgraph*.”

- And
  - So
    - From:
      - This,
- And
  - From
    - Statement 3,

– We see that:

$$I_{\mathrm{S}}$$

- Is:

"*a computation*"

done

- On:

$$V(G).$$

- And
  - So
    - ▹ Let:

$$q^*$$

- Be:

"*a non-zero sequence*"

of

- Quintuples
  - Between:

$$I_{\mathrm{S}} \qquad \textit{and} \qquad {}^{0}Q_{\mathrm{S}}.$$

- Then
  - From
    - ▹ Statement 3,
      - – We see that:

$$q^*$$

- Is:

“*a computation*”

- On:

“*a part*”

- Of:

$$G.$$

- Then
    - Since:

$$M$$

- Is:

“*deterministic*,”

- And
    - So
        - ▷ Has
            - Some:

“*precise rules*”

to

- Do
    - All:
        - ▷ things,

we see that,

- If
    - It

▷ Performs:

"*a computation*"

on

- Some
  - Part
    - ▷ Of:

$$G,$$

then

- It
  - Will:
    - ▷ Have

a

- Reason
  - *Not*
    - ▷ To
      - Perform:

"*computations*"

on

- Other
  - Parts
    - ▷ Of:

$$G.$$

- And

- So
  - From:
    - This,

- And
  - Since:

$$q^*$$

is

- Done
  - Before:

$$^0Q\text{s},$$

- We see that:

$$q^*$$

should

- Give
  - A reason
    - *Not*
      - To *perform*:

"*computations*"

on

- Some
  - Parts
    - Of:

$$G$$

which

- Does
  - *Not*
    - Contain:

$$G_c.$$

- Then
  - Since
    - Statement 1:
      - Says,

that:

$$q^*$$

can

- Only
  - Say
    - Something:
      - About

the

- Subgraph
  - Associated
    - With:
      - It,
- And
  - Since:

$$q^*$$

should

- Give
    - A reason
        - ▹ We mentioned
            - Earlier,

we see that:

$$q^*$$

should

- Give
    - A reason
        - ▹ For
            - *Not*:

“*performing*”

the

- Computation
    - To
        - ▹ Generate:

$$^0Q_\mathrm{S}$$

on

- The subgraph
    - Associated
        - ▹ With:
            - It.
- Then

- ◦ Since
  - ▹ The
    - – Given:

$$\text{“}\textit{reason}\text{”}$$

says

- That:

$$^0Q\text{s}$$

cannot

- Be
  - ◦ Generated:
    - ▹ From

the

- Subgraph
  - ◦ Associated
    - ▹ With:
      - – It,

we see that:

$$\Phi$$

should

- Have
  - ◦ Been
    - ▹ Applied:
      - – On

that:

“*part of* G.”

- Also
  - From:
    - These,

we see that,

- If
  - The
    - Number
      - Of:

“*vertices*,”

in

- That
  - Part
    - Of:

$$G,$$

which

- Corresponds
  - To:

$$q^*,$$

is

- Greater
  - Or
    - Less
      - Than:

$$n//2,$$

then

- The
  - Application
    - Of:

$$\Phi$$

will

- Imply
  - That
    - That:
      - Part

with

- A different
  - Number
    - Of:
      - Vertices

is

- Or
  - Is
    - *Not*:
      - A clique,

which

- Inturn
  - Will
    - *Not*:

– Give

the

- Required:

"*reason.*"

- And
  - So
    - From:
      - This,
- And
  - From
    - Assumption 3:
      - We see that:

$$M$$

will

- Only
  - Choose
    - Subgraphs
      - With:

"$n//2$ *vertices.*"

- And
  - So:

$$q^* \quad \Rightarrow \quad G_b \subset G \quad \textit{and} \quad \Phi(G_b) \; = \; false,$$

or

$$q^* \quad \Rightarrow \quad G_b \subset G \quad \textit{and} \quad \Phi(G_b) \; = \; true \text{ or } false,$$

such that

- The
    - Number
        - Of:
            - Vertices

in:

$$G_b$$

- Is:

$$n//2.$$

- And
    - So:

$$M$$

will

- Only
    - Choose
        - Subgraphs
            - With:

$$\text{“}n//2 \quad \textit{vertices.}\text{”}$$

- And
    - So
        - Using:
            - Lemma 1,

“*a subgraph*”

- Of:

$$G$$

that

- Does
  - *Not*
    - ▹ Satisfy:

$$\Phi$$

can

- Be
  - Deduced
    - ▹ From:

$$Q_S.$$

- Also
  - From
    - ▹ These,
      - – We see that:

$$M$$

does

- *Not*
  - Need
    - ▹ The:
      - – Result

of

- Any
  - Function

- ▹ Other
  - Than:

$$\Phi$$

after

- Choosing:

"*a subgraph.*"

- And
  - So
    - ▹ From:
      - This,
- And
  - From:
    - ▹ Assumption 3,
      - We see that:

$$M$$

will

- *Not*
  - Apply
    - ▹ Any:
      - Function

other

- Than:

$$\Phi$$

- On:

$$“\textit{a subgraph}”$$

- Of:

$$G.$$

- Then
  - Since:

$$n//2$$

depends

- On:

$$G,$$

- And
  - From:
    - Statement 1,

we see that,

- The
  - Value
    - Of:

$$n//2$$

should

- Be:

$$“\textit{calculated.}”$$

- And
  - So

        - ▷ From:
            - This,

- And
    - Since:

$$M$$

will

- Only
    - Choose
        - ▷ Subgraphs
            - With:

$$\text{“}n//2 \quad \textit{vertices,}\text{”}$$

- And
    - From:
        - ▷ Statement 1,

we see that,

- The
    - Value
        - ▷ Of:

$$n//2$$

should

- Be:

$$\text{“}\textit{calculated}\text{”}$$

before

- Choosing
    - A subgraph
        - Of:

$$G.$$

- And
    - So
        - From:
            - This,
- And
    - From
        - The:
            - Definition

of:

$$I_{\mathrm{S}},$$

- We see that:

$$n//2$$

will

- Be
    - Calculated
        - In:

$$I_{\mathrm{S}}.$$

- Then
    - We see that:

$$n//2,$$

- And
    - Criterion
        - ▹ To:
            - Choose:

“*a subgraph*”

is

- Enough
    - To
        - ▹ Choose:

“*a subgraph.*”

- And
    - So
        - ▹ If:
            - We

have

“*anything*”

more

- Than:

“*these,*”

then

- They
    - Need
        - ▹ *Not*

- Be:

"*used*"

to

- Choose

"*a subgraph.*"

- And
  - So
    - From:
      - Assumption 3,

we see that,

- Nothing
  - Other
    - These:
      - Will

be:

"*computed.*"

- And
  - So
    - From:
      - This,
- And
  - Since
    - We:
      - Saw

that,

- Computation
    - On:

$$V(G)$$

will

- *Not*
    - Give:
        - A criterion

to

- Choose:

"*a subgraph*,"

we see that,

- Nothing
    - Other
        - Than:
            - *Calculating*

the

- Value
    - Of:

$$n//2$$

will

- Be
    - Done

▷ In:

$$I_{\mathrm{S}}.$$

1^{st}NON-CLIQUE-LEMMA. *If:*

$$Q_S$$

contains

- *At least*
    - *One:*
        - ▷ *Quintuple*

that

- *Is*
    - *Not*
        - ▷ *A part*
            - *Of:*

$$I_S \quad \text{or} \quad {}^0Q_S,$$

then

- *It*
    - *Is:*
        - ▷ *Possible*

to

- *Deduce*
    - *One*
        - ▷ *Subgraph*
            - *Of:*

$$G$$

that

- *Does*
    - *Not*
        - ▹ *Satisfy:*

$$\Phi$$

- *From:*

$$Q_S.$$

So

- We see that,
    - From:
        - ▹ 1^{st}NON-CLIQUE-LEMMA,
- And
    - From:
        - ▹ Lemma 1,

we

- Are
    - In:
        - ▹ A position

to

- Define:

$$\text{“}{}^{1}Q_{\mathrm{S}} \quad (\textit{conditionally})\text{.”}$$

- And
    - So

▷ We
  - Define:

$$^{1}Q_{\mathrm{S}}$$

as:

- The
  - ○ Minimal:
    - ▷ Last
      - Part

of:

$$Q_{\mathrm{S}}$$

from

- Which:

$$G_c$$

- And: 1
  - ○ Subgraphs
    - ▷ Of:

$$G$$

that

- Does
  - ○ *Not*
    - ▷ Satisfy:

$$\Phi$$

can

- Be:

"*deduced.*"

- Then
  - In
    - ▹ The same:
      - Way

we

- Proved:
  - 1^{st}NON-CLIQUE-LEMMA,

we

- Can
  - Also:
    - ▹ State,
- And
  - Prove:
    - ▹ 2^{nd}NON-CLIQUE-LEMMA.

2^{nd}NON-CLIQUE-LEMMA

- Says:
  - That,
    - ▹ If:

$$Q_{\mathrm{S}}$$

contains

- At least
  - *One*:

▹ Quintuple

that

- Is
  - *Not*
    - ▹ A part
      - Of:

$$I_{\mathrm{S}} \quad \textit{or} \quad {}^{1}Q_{\mathrm{S}},$$

then

- It
  - Is:
    - ▹ Possible

to

- Deduce
  - *One*
    - ▹ Subgraph
      - Of:

$$G$$

that

- Does
  - *Not*
    - ▹ Satisfy:

$$\Phi$$

- From:

$$Q_{\mathrm{S}}.$$

- And
    - So
        - ▹ Using:
            - 2^{nd}NON-CLIQUE-LEMMA

we

- Can
    - Define:

$$\text{“}{}^{2}Q_{\mathrm{S}} \quad (\textit{conditionally})\text{.”}$$

- And
    - Also
        - ▹ We
            - Can:

"*continue this*"

a

- Finite
    - Number
        - ▹ Of:
            - Times.
- And
    - So:
        - ▹ We

can

- Define:
    - 3^{rd}NON-CLIQUE-LEMMA, ${}^{3}Q_{\mathrm{S}}$,

- $\circ$ 4^{th}Non-Clique-Lemma, ${}^{4}Q_{\mathrm{S}}$,

  $\vdots$
- $\circ$ j^{th}Non-Clique-Lemma, ${}^{j}Q_{\mathrm{S}}$.

- Then we see that,
  - At
    - ▷ This:
      - Point,

we

- Do
  - *Not*:
    - ▷ Know

what

- Values:

$$j$$

- Can:

"*have.*"

- Or
  - We see that,
    - ▷ At:
      - Present,

we

- Know
  - That
    - ▷ The:

- Value

of:

$$j$$

is

- At
  - Least:

$$0,$$

- And
  - That
    - Is:
      - All.
- Then
  - If
    - It:
      - Is

the

- Case
  - That,
    - If:

$$j$$

is

- Always
  - Equal
    - To:

$$0,$$

- Then:

$$M$$

will

- Always
  - Perform:

$$I_S,$$

- And
  - *Immediately*
    - ▹ Choose:

$$G_c.$$

- But
  - If
    - ▹ We
      - Can:

“*prove*”

- That:

$$j$$

can

- Have
  - A value
    - ▹ Greater
      - Than:

$$0,$$

then

- In
    - Some
        - ▹ Cases:

$$M$$

will

- Choose
    - Some:

“*subgraphs*,”

which

- Will
    - *Not*
        - ▹ Satisfy:

$$\Phi$$

before

- Choosing:

$$G_c.$$

- And
    - So
        - ▹ By:
            - That,

we

- Can
    - Prove:
        - That,

$$M$$

does

- Indeed
    - Have
        - A particular:
            - Structure.

Definition 8. MACHINE-STRUCTURE

- Is
    - Defined
        - As:

begin

- Let:

$$G_b$$

- Be:

"*a subgraph*"

- Of:

$$G$$

that

- Has
    - *Not*

▷ Yet

– Been:

"*considered.*"

- If:

$$\Phi(G_b) \quad = \quad true,$$

- Then:

"*halt,*"

- And
  - *Accept*:

$$G.$$

- Else
  - If:

$$\Phi(G_b) \quad = \quad false,$$

- And
  - There

▷ Are:

"*subgraphs*"

still

- Left
  - To

▷ Be:

"*considered,*"

then

- *Recursively*
  - Call
    - This:
      - *Procedure.*
- Else
  - If
    - *All*:

"*subgraphs*"

have

- Been:

"*considered,*"

- Then:

"*halt,*"

- And
  - *Reject*:

$$G.$$

end.

Let:

$$G_1 \quad \textit{and} \quad G_2$$

be

- *Two*
  - Graph

▷ With:

$$\text{“}n \quad \textit{vertices,”}$$

- And
  - Let:
    - ▷ $E(G_1) \quad = \quad \mathrm{S}_1,$
    - ▷ $E(G_2) \quad = \quad \mathrm{S}_2.$
- And
  - Also
    - ▷ Let:

$$I_{\mathrm{S}_1} \qquad \textit{and} \qquad I_{\mathrm{S}_2}$$

be

- The
  - Respective
    - ▷ Computations:
      - Done

for:

$$\mathrm{S}_1 \qquad \textit{and} \qquad \mathrm{S}_2.$$

- Then
  - Since
    - ▷ Both:

$$G_1 \qquad \textit{and} \qquad G_2$$

have

- The
  - Same

- ▷ Number
  - – Of:

"*vertices*,"

- And
  - Since:

$$\mathrm{S}_1 \qquad \textit{and} \qquad \mathrm{S}_2$$

are

- Both
  - Constructed
    - ▷ Using:

$$E,$$

- And
  - Since:

$$I_{\mathrm{S}_1} \qquad \textit{and} \qquad I_{\mathrm{S}_2}$$

are

- Both
  - Performed
    - ▷ By:

$$M,$$

- And
  - Since:

$$I_{\mathrm{S}_1} \qquad \textit{and} \qquad I_{\mathrm{S}_2}$$

are

- Done
  - To:
    - ▹ Calculate

the

- Value
  - Of:

$$n//2,$$

- We see that:

$$I_{\mathrm{S}_1} \quad \textit{and} \quad I_{\mathrm{S}_2}$$

are

- The
  - *Same*:
    - ▹ Computations.

Lemma 3. *Let:*

$$G_1 \quad \textit{and} \quad G_2$$

be

- *Two*
  - *Graph*
    - ▹ *With:*

$$\text{“}n \quad \textit{vertices,”}$$

- *And*
  - *Let:*
    - ▹ $E(G_1) \;=\; S_1,$
    - ▹ $E(G_2) \;=\; S_2.$

- *Then:*

$$I_{S_1} \quad \textit{and} \quad I_{S_2}$$

are

- *The*
  - *Same:*
    - ▷ *Computations.*

Assume

- That:

$$G_1, \quad \ldots, \quad G_{j-1}, \quad G_j, \quad \ldots, \quad G_m$$

are

- The
  - *Only*
    - ▷ Graphs
      - With:

$$\text{“}n \quad \textit{vertices,}\text{”}$$

- Such that:

$$E(G_1) \quad = \quad \mathrm{S}_1, \quad \ldots, \quad E(G_{j-1}) \quad = \quad \mathrm{S}_{j-1},$$
$$E(G_j) \quad = \quad \mathrm{S}_j, \quad \ldots, \quad E(G_m) \quad = \quad \mathrm{S}_m,$$

$$\textit{and}$$

$$M(\mathrm{S}_1) \quad = \quad 1, \quad \ldots, \quad M(\mathrm{S}_{j-1}) \quad = \quad 1,$$
$$M(\mathrm{S}_j) \quad = \quad 1, \quad \ldots, \quad M(\mathrm{S}_m) \quad = \quad 1.$$

- Then
  - Since

- ▹ All
  - These:

"*strings*"

- Are:

"*different*,"

- And
  - Since:

$$M$$

- Behaves:

"*differently*"

for

- *Different*
  - Strings,
    - ▹ Assume:

$$|Q_{\mathrm{S}_1}| \leqslant \quad \ldots \quad \leqslant |Q_{\mathrm{S}_{j-1}}| \leqslant |Q_{\mathrm{S}_j}| \leqslant \quad \ldots \quad \leqslant |Q_{\mathrm{S}_m}|.$$

- But
  - Since:
    - ▹ For
      - All: S_j

$$M(\mathrm{S}_j) \quad = \quad 1,$$

- Assume:

$$G_{c_{j-1}} \subset G_{j-1} \qquad \textit{and} \qquad G_{c_j} \subset G_j$$

- Such that:

$$\text{“}\textit{both} \quad G_{c_{j-1}} \quad \textit{and} \quad G_{c_j} \quad \textit{satisfies} \quad \Phi,\text{”}$$

- And
  - So
    - ▹ We:
      - Can

say

- That:

$$Q_{\mathrm{S}_j} \quad \textit{ends with} \quad {}^0Q_{\mathrm{S}_j},$$

$$Q_{\mathrm{S}_{j-1}} \quad \textit{ends with} \quad {}^0Q_{\mathrm{S}_{j-1}},$$

- And
  - So:

$$\mathrm{S}_j$$

- Contains:

“*a sub string,*”

- Say:

$$P_{\mathrm{S}_j}$$

- Such that:

$$\text{“}P_{\mathrm{S}_j} \quad \textit{is a part of} \quad \mathrm{S}_j$$

only because

$$G_{c_j} \quad \textit{is a subgraph of} \quad G_j\text{”}$$

- And
  - Similarly,

▹ For:

$$P_{\mathrm{S}_{j-1}}.$$

- But

  ◦ We see that,

    ▹ In

      – General:

$$\text{“}\textit{the part}\text{”}$$

of

- The

  ◦ String:

$$\mathrm{S}_{j-1}$$

before

- And

  ◦ *After*

    ▹ The:

      – Occurrence

of

- The

  ◦ Sub string:

$$P_{\mathrm{S}_{j-1}}$$

will

- Be

  ◦ Different

- ▹ From
  - The:

“*the part*”

of

- The
  - String:

$$S_{j-1}$$

before

- And
  - *After*
    - ▹ The:
      - Occurrence

of

- The
  - Sub string:

$$P_{S_j}.$$

- Then
  - Since:

$$|Q_{S_{j-1}}| \leqslant |Q_{S_j}|,$$

- And
  - Since:

$$E$$

is

- Used
  - To
    - ▹ Construct
      - – Both:

$$\mathrm{S}_{j-1} \qquad \textit{and} \qquad \mathrm{S}_j,$$

- And
  - Since
    - ▹ Lemma 3:
      - – Says

that,

- The
  - Computations
    - ▹ For:

$$I_{\mathrm{S}_{j-1}} \quad \textit{and} \quad I_{\mathrm{S}_j}$$

are

- The:

“*same*,”

we see that,

- After
  - Performing:

$$I_{\mathrm{S}_{j-1}} \qquad \textit{and} \qquad I_{\mathrm{S}_j}$$

- On:

$$\mathrm{S}_{j-1} \quad \textit{and} \quad \mathrm{S}_j,$$

the

- Head
    - Of:

$$M$$

will

- Point
    - To
        - An:
            - *Alphabet*

that

- Is
    - A part
        - Of:

$$P_{\mathrm{s}_{j-1}},$$

- And
    - The
        - Head
            - Of:

$$M$$

will

- *Not*
    - Point
        - To:
            - An *alphabet*

that

- Is
    - A part
        - Of:

$$P_{\mathrm{s}_j}.$$

- And
    - Similarly,
        - In:
            - Another case

the

- Head
    - Of:

$$M$$

will

- *Not*
    - Point
        - To:
            - An *alphabet*

of

- Either:

$$P_{\mathrm{s}_{j-1}} \quad \textit{or} \quad P_{\mathrm{s}_j}.$$

- And
    - So
        - From:

- This,

we see that,

- There
  - Will:
    - *Exist*

at

- Least:
  - *One*
    - Quintuple
      - In:

$$Q_{\mathrm{S}_j}$$

that

- Will
  - *Not*
    - Be:

"*a part*"

- Of:

$$I_{\mathrm{S}_j} \qquad \textit{or} \qquad {}^{0}Q_{\mathrm{S}_j}.$$

- And
  - So
    - From:
      - 1^{st}NON-CLIQUE-LEMMA,

we see that:

$${}^{1}Q_{\mathrm{S}_j}$$

will

- Be:

$$\text{“}a\ part\text{”}$$

- Of:

$$Q_{s_j}.$$

- And
    - So:
        - Iterating

this

- Way,
    - We see that:

$$^{j}Q_{s_m}$$

will

- Be:

$$\text{“}a\ part\text{”}$$

- Of:

$$Q_{s_m},$$

such that

- The
    - Maximum
        - Value
            - Of:

$$j$$

- Is:

$$\text{“}\mathit{unknown}\text{,”}$$

- And
  - Also:

$$j$$

may

- Or
  - May:
    - *Not*

be

- Equal
  - To:

$$m.$$

Lemma 4. *When:*

$$M$$

- *Accepts:*

$$\text{“}\mathit{a\ string,}\text{”}$$

in

- *Some:*
  - *Cases,*

it

- *Will*

- *Choose*
  - *A certain:*
    - *Number*

of:

$$\text{“non cliques”}$$

before

- *Choosing*
  - *A clique*
    - *Of*
      - *Size:*

$$n//2.$$

Note that,

- In
  - The
    - Strict:
      - Sense,

we

- Should
  - First
    - Prove:
      - 1^{st}NON-CLIQUE-LEMMA,
  - And
    - Then:
      - 1^{st}NON-CLIQUE-EXISTENCE-LEMMA.
- And

- Then
  - ▹ Move
    - – A step:

"*backwards*,"

- And
  - Prove:
    - ▹ 2^{nd}NON-CLIQUE-LEMMA,
      - – And 2^{nd}NON-CLIQUE-EXISTENCE-LEMMA

⋮

- And
  - So
    - ▹ From:
      - – This,

we see that,

- For
  - The
    - ▹ Case
      - – When:

$$M$$

- Accepts:

"*a string*,"

it

- Will
  - Have:

"a structure,"

- And
    - The:

"definition"

of

- That:

"structure"

will

- Be
    - To
        - *Choose*
            - Subgraphs:

"one by one."

- Then
    - Since
        - That:
            - Definition

is

- To
    - Be
        - Always:
            - That way,

we see that,

- We

- Cannot
  - ▷ Remove
    - Any:

"*rules*"

from

- That:

"*definition.*"

- Also
  - If we
    - ▷ Add
      - Some:

"*rules*"

into

- That:

"*definition,*"

then

- Also
  - That:

"*definition,*"

will

- Still be
  - To
    - ▷ *Choose*
      - Subgraphs:

“one by one.”

- And
  - So
    - ▹ We
      - Cannot:

add *or* *remove*

- Any:

“rules”

into

- Or
  - From
    - ▹ That:

“definition.”

- But
  - We see that:

$$M$$

cannot

- Know
  - Whether:

“a string”

will

- Be:

accepted *or* *rejected*

unless

- It
  - Does
    - *Some*:
      - Computation.
- And
  - So
    - Assume
      - That:

$$M$$

has

- *Two*:
  - Kinds

of

- Structures
  - In
    - It:
- One
  - For:
    - The
      - Case

when

- The
  - Given
    - String:

- Will

be:

“*accepted,*”

- And
  - The:
    - Other

for

- The
  - Case:
    - When

the

- Given
  - String
    - Will
      - Be:

“*rejected.*”

- And
  - So
    - Let:

$$\mathscr{A}_M \qquad \text{and} \qquad \mathscr{R}_M$$

be

- Those:

“*structures*”

- In:

$$M$$

for

- The *accept*
  - And:
    - *Reject*
      - Cases.
- And
  - So
    - Before:

$$M$$

starts

- The
  - Actual:
    - Computation,

it

- Will
  - Make
    - *Two*:
      - Copies

of

- The
  - Given:

"*string*,"

- And
    - Send:

"a copy"

to

- Each
    - Of
        - Those:

"structures."

- And
    - After
        - Those:

"two structures"

have

- Done
    - Their
        - Respective:

"computations,"

they

- Will
    - Pass
        - On
            - Their:

"results"

- To:

$$M,$$

- And:

$$M$$

will

- Announce
  - The
    - Result:
      - Accordingly.
- But
  - We:
    - Who

are

- Standing
  - Outside
    - Will:
      - Know

whether

- A particular:

"*string*"

is

- Going
  - To
    - Be:

accepted *or* *rejected.*

- And
  - So
    - ▹ Assume:
      - That,

we

- Have
  - Given:

"*a string*"

which

- We
  - Know:
    - ▹ That

it

- Will
  - Be:

"*accepted*"

- By:

$$M.$$

- Then
  - Since:

$$M$$

does

- *Not*
  - Know
    - Whether
      - This:

"*string*"

is

- Going
  - To
    - Be:

accepted or *rejected*

unless

- It
  - Performs
    - *Some*:
      - Computation,

we see that:

$$M$$

will

- Make
  - *Two*
    - Copies:
      - Of it,
- And
  - Send:
    - Them

to

- Each
    - Of
        - ▹ Those
            - Two:

"*structures.*"

- Then
    - We see that,
        - ▹ Both:

$$\mathscr{A}_M \qquad \textit{and} \qquad \mathscr{R}_M$$

will

- Say
    - That,
        - ▹ The
            - Given:

"*graph*"

- Contains:

"*a clique.*"

- And
    - Similarly,
        - ▹ When we
            - Give:

"*a string*"

which

- We
  - Know:
    - That

it

- Will
  - Be:

$$\text{“}\textit{rejected}\text{”}$$

- By:

$$M,$$

then

- Both:

$$\mathscr{A}_M \qquad \textit{and} \qquad \mathscr{R}_M$$

will

- Say
  - That,
    - The:

$$\text{“}\textit{graph}\text{”}$$

does

- *Not*
  - Contain:

$$\text{“}\textit{a clique.}\text{”}$$

- And

- So
  - From:
    - This,

we see that,

- At
  - The
    - Same:
      - Time,

the

- Structure:

$$\mathscr{R}_M$$

can

- Act
  - As:

"*a structure*"

for

- *Accept*
  - And
    - *Reject*:
      - Cases.
- And
  - So
    - From:
      - This,

- And
    - Since
        - There:
            - Cannot

be

- Any
    - Other:

$$\text{“}\textit{rules}\text{”}$$

in

- The
    - *Definition*
        - *For*:

$$\mathscr{A}_M,$$

we see that,

- There
    - Will
        - Be:

$$\text{“}\textit{no other rules}\text{”}$$

in

- The
    - Definition
        - *For*:

$$\mathscr{R}_M$$

other

- Than
    - That
        - In:

$$\mathscr{A}_M.$$

- And
    - So
        - The:

“*definition*”

- For:

$$\mathscr{R}_M$$

will

- Be
    - The:
        - *Same*

as

- That
    - Of:

$$\mathscr{A}_M.$$

Theorem 2.1. M

- *Has:*

“*a* MACHINE-STRUCTURE.”

Note that,

- Since
  - Statement 3,
    - Says
      - That:

$$M$$

should

- Work
  - On
    - The
      - Given:

"*graph*,"

- And
  - Since
    - The:
      - Result

of:

$$E$$

is

- The
  - Only
    - Thing:
      - Given

to:

$$M,$$

- We see that:

$$E$$

should

- Be:
  - A bijective
    - Function,
- And
  - Also
    - The:
      - Result

of:

$$E$$

will

- Be:

“*a representation*”

of

- The
  - Given:

“*graph*.”

Lemma 5. $E(G)$

- *Is:*

“a representation”

- *Of:*

$$G.$$

3 Machine combinatorics

Definition 9. We

- Define:
    - A variable
        - ▷ As:
            - Something

that

- Can
    - Store
        - ▷ An arbitrary:
            - Element

from

- The set
    - Of
        - ▷ All:
            - Integers.

So we see that,

- If
    - We:
        - ▷ Note
            - Down

the

- Sequence
    - Of:
        - ▷ Elements

- Stored:

in:

$$\text{“}\textit{a variable,}\text{”}$$

then

- We
  - Will
    - Get:

$$\text{“}\textit{an inductive sequence.}\text{”}$$

- And
  - So
    - We can:
      - Consider

a

- Variable
  - As
    - Something:
      - Equivalent

to

- An
  - Inductive:
    - Sequence.
- Let:

$$f$$

be

- A deterministic:

“*function.*”

- Then
  - If:

$$f(a_{10}) \quad = \quad a_{11},$$

we see that,

- The
  - Value
    - ▷ Returned
      - By:

$$f$$

will

- Be:

$$a_{11}$$

if

- And
  - Only
    - ▷ If

the

- Parameter
  - Given
    - ▷ To:

$$f$$

- Is:

$$a_{10}.$$

- Or
  - Informally,
    - If

the

- Parameter
  - Given
    - To:

$$f$$

is

- *Not*:

$$a_{10},$$

then

- The
  - Value:
    - Returned
      - By:

$$f$$

will

- *Never*
  - Be:

$$a_{11}.$$

- And
    - So:
        - When

a

- Function
    - Receives:
        - A parameter,
- And
    - Returns:
        - Something,

then

- That
    - Value:
        - Returned

can

- Only
    - Be:
        - Gotten

by

- Making
    - Some:
        - Changes

in

- The:

“*parameter.*”

- But
    - We see that:

$$f$$

always

- Remains:
    - The
        - Same.
- And
    - So:
        - There

are

- Some:
    - *Fixed*
        - And *finite*
            - Number

of

- Rules
    - In
        - The:
            - Definition

of:

$$f.$$

- And

- So
  - ▹ At:
    - Anytime,

the

- Value
  - Returned
    - ▹ By:

$$f$$

can

- Only
  - Be:
    - ▹ Gotten

by

- Making
  - *One*
    - ▹ Among:

“w *changes*”

in

- The:

“*parameter.*”

- But
  - When
    - ▹ The change:
      - Made

can

- Be
  - *One*
    - Among:

“w *changes*,”

we see that,

- If
  - Only:

“$w - 1$”

among

- All
  - These:

“w *changes*,”

will

- Ever
  - Be:
    - *Made*,

then

- There
  - Will
    - Be:
      - *No* point

in

- Saying
    - That:

$$f$$

can

- Make
    - *One*
        - Among:

"w *changes*."

- And
    - So:
        - *One*
            - Among

all

- Those:

"w *changes*"

will

- Be:

"*made*"

in

- At
    - Least:
        - *One*
            - Case.

Lemma 6. *When*

- *A function*
  - *Receives:*
    - ▹ *A* parameter,
- *And*
  - *Returns:*
    - ▹ *A* value,

then

- *That*
  - *Value:*
    - ▹ *Returned*

can

- *Be:*
  - *Obtained*

only

- *By*
  - *Making:*

“w changes”

- *In:*

“the parameter.”

- *And*
  - *Also:*
    - ▹ One
      - *Among*

all

- *Those:*

$$\text{“}w \quad \text{changes”}$$

will

- *Be:*

“made”

in

- *At least*
  - *One*
    - *Case.*

3.1 Finite machine combinatorics

Let:

$$M$$

- Be:

“*a Turing machine*”

- Which:

“*halts*”

for

- All:

“*strings.*”

- And
  - Also

- Assume
  - That,

the

- Program:

$$\chi(M)$$

will

- *Never*:

"*change*,"

- And
  - Also
    - That,

the

- Machine:

$$M$$

- Always:

"*starts*"

from

- The
  - Left
    - Most:
      - *Alphabet*

of

- The

- ◦ Given:

$$\text{“}string\text{.”}$$

- Then
  - Since
    - ▷ The:
      - Computation

done

- By:

$$M$$

is

- A step
  - By
    - ▷ Step:
      - Process,

we see that,

- The
  - Computation
    - ▷ Done
      - By:

$$M$$

will

- Be
  - An infinite:
    - ▷ Sequence

- Of:

"*integers.*"

- And
  - So:

$$M$$

can

- Be
  - Considered
    - As
      - Something:

"*equivalent*"

to:

- An
  - Inductive
    - Sequence.
- And
  - So
    - All:
      - Sequences

in:

$$M$$

will

- Correspond
  - To:

        - A variable.

- And
    - So:

$$M$$

will

- Have
    - At least:
        - One
        - Or some
            - Variables in it.

- Then
    - Since:

$$M$$

has:

- *Only*
    - An initial state, $\chi(M)$,
        - And some:
            - Terminal states,

- And
    - Since:

$$\chi(M)$$

will

- *Never*:

“*change*”

- And
  - Since:

$$\chi(M)$$

is

- Equivalent
  - To
    - An inductive:
      - Sequence,

we see that,

- The
  - Set:

$$\chi(M)$$

can

- Be:

"*divided*"

in

- Such
  - A way
    - That,
      - Each:

"*division*"

will

- Correspond

- To:

$$\text{“}\textit{a variable}\text{”}$$

- Of:

$$M.$$

- And
  - Also
    - Each:
      - *Division*

will

- Correspond
  - To:
    - The
      - Function

that

- Generates
  - The
    - Next

in

- An
  - Inductive:
    - Sequence
      - Of:

$$M.$$

- Then

- ◦ Since
  - ▹ Whole
    - – Set:

$$\chi(M)$$

is

- A part
  - ◦ Of
    - ▹ The
      - – Same:

"*definition*"

- Of:

$$M,$$

we see that,

- The
  - ◦ Set:

$$\chi(M)$$

cannot

- Be:

"*divided*"

into

- Independent:

"*entities.*"

- And

- So:
  - All

those

- Sequences:

“*corresponding*”

to

- All
  - Those:
    - Variables

will

- Be:

“*intertwined*,”

- And
  - So
    - If:
      - There

are:

“2 *variables*”

- In:

$$M,$$

then

- The
  - Computation:

- ▹ Done
  - – By:

$$M$$

will

- Be:
  - ○ A sequence
    - ▹ Of:

"2-*tuples*."

- Also
  - ○ Since:

$$\chi(M)$$

will

- *Never*:

"*change*"

we see that,

- The
  - ○ Number
    - ▹ Of:
      - – Variables

in

- The
  - ○ Machine:

$$M$$

will

- *Not*:

"*increase.*"

- Also
  - If:
    - The
      - Number

of

- Variables
  - Decreases
    - By:
      - *One*,

then

- That
  - Thing
    - Which

was

- Previously
  - Called:
    - A variable;

will

- *Not*
  - Generate
    - An infinite:
      - Sequence.

- And
    - So
        - ▷ That:
            - Thing

which

- Was
    - Called:
        - ▷ A variable;

will

- *Not*
    - Satisfy
        - ▷ The definition
            - Of:

$$\text{“}\textit{a variable.}\text{”}$$

- And
    - So
        - ▷ The:
            - Number

of

- Variables
    - In:

$$M,$$

will

- *Never*:

“*decrease.*”

- And
  - So
    - ▹ Assume:
      - That,

there

- Are
  - *Two*:
    - ▹ Variables
      - In:

$$M.$$

- Then
  - The inductive
    - ▹ Sequence
      - In:

$$M$$

can

- Be
  - Written
    - ▹ As:

$$(a_0,\ \ b_0), \qquad (a_1,\ \ b_1), \qquad (a_2,\ \ b_2), \qquad \ldots.$$

- Then since
  - Each:
    - ▹ Division

- Of:

$$\chi(M)$$

corresponds

- To
    - The:
        - ▷ Function

that

- Generates
    - The next
        - ▷ Integer:
            - Stored

in:

“*a variable*,”

assume

- That:

$$\mathscr{V}_a \qquad \textit{and} \qquad \mathscr{V}_b$$

are

- Those
    - *Two*:
        - ▷ Divisions
            - Of:

$$\chi(M).$$

- Then
    - When

▷ We are:
  – Going

to

- Generate:

  ○ The:

$$\text{“}10^{th} \quad \textit{element}\text{”}$$

we see that,

- One

  ○ Way

    ▷ In:
      – Which

we

- Can

  ○ Generate:

    ▷ The:

$$\text{“}10^{th} \quad \textit{element}\text{”}$$

is,

- Choose:

  ○ a_8 as the first parameter of: $\mathscr{V}_a$,
  ○ a_9 as the second,
  ○ b_8 as the third,
  ○ b_9 as the fourth,

- And choose:

  ○ b_8 as the first parameter of: $\mathscr{V}_a$,

- b_9 as the second,
- a_8 as the third,
- a_9 as the fourth.

- And
  - The:

$$\text{“}10^{th} \quad \textit{element}\text{”}$$

in

- The
  - Sequence
    - Will
      - Be:

$$(\mathscr{V}_a(a_8,\ a_9,\ b_8,\ b_9), \quad \mathscr{V}_b(b_8,\ b_9,\ a_8,\ a_9)).$$

- And
  - So
    - Iterating,

we see that,

- The:

$$\text{“}11^{th} \quad \textit{element}\text{”}$$

will

- Be:

$$(\mathscr{V}_a(a_9,\ a_{10},\ b_9,\ b_{10}), \quad \mathscr{V}_b(b_9,\ b_{10},\ a_9,\ a_{10})).$$

Therefore

- Since:

$$\chi(M)$$

will

- *Never*:

$$\text{“\textit{change},”}$$

- And
    - Since:

$$\chi(M)$$

- Is:

$$\text{“\textit{always}”}$$

used

- To
    - Perform
        - ▷ All:

$$\text{“\textit{computations},”}$$

we see that,

- If
    - The:

$$\text{“}10^{th} \quad \textit{element}\text{”}$$

is

$$(\mathscr{V}_a(a_8,\ \ a_9,\ \ b_8,\ \ b_9), \qquad \mathscr{V}_b(b_8,\ \ b_9,\ \ a_8,\ \ a_9)),$$

then

- The:

$$\text{“}11^{th} \quad \textit{element}\text{”}$$

will

- Always
  - Be:
    - $\triangleright$ $(\mathscr{V}_a(a_9,\ a_{10},\ b_9,\ b_{10}),\quad \mathscr{V}_b(b_9,\ b_{10},\ a_9,\ a_{10})),$
- And
  - Never:
    - $\triangleright$ $(\mathscr{V}_a(a_6,\ a_7,\ b_5,\ b_{10}),\quad \mathscr{V}_b(b_8,\ b_{10},\ a_8,\ a_{10})),$
    - $\triangleright$ $(\mathscr{V}_a(a_5,\ a_{10},\ b_9,\ b_{10}),\quad \mathscr{V}_b(b_8,\ b_{10},\ a_7,\ a_{10},\ b_7)).$
- And
  - The:

$$\text{“}12^{th} \quad \textit{element}\text{”}$$

will

- Always
  - Be:
    - $\triangleright$ $(\mathscr{V}_a(a_{10},\ a_{11},\ b_{10},\ b_{11}),\quad \mathscr{V}_b(b_{10},\ b_{11},\ a_{10},\ a_{11})),$
- And
  - Never:
    - $\triangleright$ $(\mathscr{V}_a(a_{10},\ a_5,\ b_4,\ b_{11},\ b_3),\quad \mathscr{V}_b(b_8,\ b_{10},\ a_8,\ a_{10})),$
    - $\triangleright$ $(\mathscr{V}_b(a_7,\ a_{10},\ b_4,\ b_{10}),\quad \mathscr{V}_a(b_8,\ b_{10},\ a_7,\ a_{11},\ b_7)).$

Therefore

- Since:

$$\chi(M)$$

will

- *Never*:

"*change*,"

and

- Since:

$$\chi(M)$$

is

- *Always*
  - Used
    - To:
      - Perform

all:

"*computations*,"

we see that,

- *Not* only
  - Will
    - The
      - Functions:

$$\mathscr{V}_a \quad \text{and} \quad \mathscr{V}_b,$$

- And
  - The number
    - Of
      - Its:

"*parameters*"

will

- Always
    - *Remain*
        - ▹ The:
            - Same,
- But
    - Also
        - ▹ There:
            - Will

be

- A *fixed*
    - Order:
        - ▹ For:
            - Choosing

the:

"*parameters*"

- For:

$$\mathscr{V}_a \quad \textit{and} \quad \mathscr{V}_b,$$

- And
    - That
        - ▹ Order

will

- Always
    - *Remain*
        - ▹ The:
            - Same

during

- The
    - Entire:

"*computation.*"

- And
    - So
        - ▹ At:
            - Anytime,

there

- Will be
    - Some:
        - ▹ *Fixed*
            - Rules

to

- *Choose*
    - Some
        - ▹ Previous:
            - Results,
- And
    - Also:
        - ▹ Only

a

- Particular:

"*computation*"

can

- Be
    - Done
        - Using
            - Those:

"previous results."

- And so
    - There
        - Will:
            - Always

be

- Some:

"rules"

to

- Perform
    - All:

"computations."

- And
    - Those:

"rules"

will

- Always
    - Remain:

- ▷ *Fixed*,
  - And *finite*,

since:

$$\chi(M)$$

is

- Always:
  - ◦ *Fixed*
    - ▷ And
      - *Finite.*

Statement 4. *If:*

$$M$$

uses

- *Some*
  - ◦ *Previous:*

"results"

to

- *Get*
  - ◦ *The*
    - ▷ *Next:*
      - *Result,*

then

- *It*
  - ◦ *Can*
    - ▷ *Only:*
      - *Use*

a

- *Finite*
    - *Number*
        - *Of:*

“previous results,”

- *And also*
    - *Only*
        - *Those*
            - *Particular:*

“previous results”

can

- *Be:*

“used,”

- *And*
    - *Also:*
        - *Only*

a

- *Particular:*

“computation”

can

- *Be*
    - *Done*
        - *Using:*
            - *Them*

to

- *Get*
  - *The*
    - *Next:*
      - *Result.*

Statement 5. *If:*

$$M$$

uses

- *Some*
  - *Previous:*
    - *Results*

to

- *Get*
  - *The*
    - *Next:*
      - *Result,*

then

- *There*
  - *Will:*
    - *Always*

be

- *A* fixed
  - *And*
    - Finite
      - *Number*

of

- *Rules*
  - *For:*
    - *Getting*
      - *It.*

But we see that,

- The
  - Next
    - Element:
      - Generated

will

- Also:

"*depend*"

on

- The
  - String
    - Given
      - To:

$$M.$$

- And
  - So
    - The
      - Given:

"*string*,"

- Or

"*a sub string*"

of

- The
  - Given:

"*string,*"

will

- Be
  - A parameter
    - ▷ For
      - Generating:

"*the next.*"

- But
  - Since:

$$\chi(M)$$

- Always:

"*dictates*"

what

- Is
  - To
    - ▷ Be:
      - Done,
- And

- Since:

$$\chi(M)$$

will

- *Never*:

"*change*,"

we see that,

- An arbitrary
  - Sub string
    - Of:
      - The

"*string*,"

on

- The
  - Tape
    - Of:

$$M$$

will

- *Not*
  - Be:

"*a parameter*"

to

- Generate
  - The:

        - ▹ Next.
- And
    - ○ Also:
        - ▹ A sub string

of

- Arbitrary
    - ○ Length
        - ▹ Will:
            - *Not*

be:

“*a parameter*”

to

- Generate
    - ○ The:
        - ▹ Next.
- But
    - ○ At
        - ▹ Each:
            - Step,

there

- Will
    - ○ Be
        - ▹ Some
            - Precise:

“*rules*”

to

- Choose
    - The:
        - ▹ Sub string

that

- Will
    - Be
        - ▹ The:

“*parameter*”

to

- Generate
    - The:
        - ▹ Next.

Statement 6. *At*

- *Each*
    - *Step,*

there

- *Will*
    - *Be*
        - ▹ *Some*
            - *Precise:*

“rules”

to

- *Choose*

- *The:*

$$\text{“\textit{sub string}”}$$

that

- *Will*
    - *Be*
        - ▹ *The:*

$$\text{“\textit{parameter}”}$$

to

- *Generate*
    - *The:*
        - ▹ *Next.*

So assume that,

- We have
    - Given:
        - ▹ A string
            - To:

$$M.$$

- Then
    - With accordance
        - ▹ To:
            - Statement 6,

we see that:

$$M.$$

will

- Choose:
  - A sub string
    - Say:

$$S_0.$$

- And
  - Also
    - There:
      - Will

be

- Some
  - Precise:
    - *Rules*
      - To choose:

$$S_i,$$

- The:

"i^{th} *sub string.*"

- And so
  - There
    - Will:
      - *Exist*

a

- Function:

$$\hbar$$

- Such that:

$$\mathrm{S}_{i-1} \quad \xrightarrow{\hbar} \quad \mathrm{S}_i.$$

- And
  - Also:
    - ▹ For

the

- Sake
  - Of:

“*simplicity*,”

assume

- That:

$$M$$

has

- *Two*
  - Variables,
    - ▹ Say $\mathscr{V}_a, \ \mathscr{V}_b$
      - In it.
- And
  - Let:

$$\mathscr{V}_a(i) \qquad \textit{and} \qquad \mathscr{V}_b(i)$$

be

- Used
  - To:
    - ▹ Denote

the

- i^{th} value
  - Stored
    - ▹ In
      - Those:

"*variables.*"

- Also
  - With accordance
    - ▹ To:
      - Statements 4 and 5,

let

- The:

"i^{th} *tuple*"

- Be:

$$(\mathscr{V}_a(\mathscr{V}_a(i-1),\ \mathscr{V}_b(i-1),\ \mathrm{S}_{i-1}),\quad \mathscr{V}_b(\mathscr{V}_a(i-2),\ \mathscr{V}_b(i-2),\ \mathrm{S}_{i-1})).$$

- But
  - Since
    - ▹ There is
      - *No*:

$$\mathscr{V}_a(-1) \qquad \textit{and} \qquad \mathscr{V}_b(-1)$$

assume that,

- There
  - Is
    - ▹ Also

- *No*:

$$\mathscr{V}_a(0) \quad \textit{and} \quad \mathscr{V}_b(0).$$

- And
  - So
    - ▷ The first
      - Tuple:

$$\text{“}\textit{generated}\text{”}$$

will

- Be:

$$\mathscr{V}_a(1) \quad \textit{and} \quad \mathscr{V}_b(1).$$

- And
  - Also
    - ▷ The first
      - Tuple:

$$\text{“}\textit{generated}\text{”}$$

will

- Depend:
  - Only
    - ▷ On:

$$\mathrm{S}_0,$$

the

- First
  - Sub string:

- ▷ Chosen.

- And
  - So
    - ▷ The:
      - Computation

to

- Generate
  - The
    - ▷ *First*:
      - Tuple

can

- Be
  - Written
    - ▷ As:
      - $\mathrm{S}_0 \xrightarrow{\mathscr{V}_a} \mathscr{V}_a(1)$,
      - $\mathrm{S}_0 \xrightarrow{\mathscr{V}_b} \mathscr{V}_b(1)$,
      - $\mathrm{S}_0 \xrightarrow{\hbar} \mathrm{S}_1$.

- And
  - The:
    - ▷ Computation

to

- Generate
  - The
    - ▷ *Second*:
      - Tuple

can

- Be
  - Written
    - ▷ As:
      - $(\mathscr{V}_a(1),\ \mathscr{V}_b(1),\ \mathrm{S}_1) \ \xrightarrow{\mathscr{V}_a}\ \mathscr{V}_a(2),$
      - $\mathrm{S}_1 \ \xrightarrow{\mathscr{V}_b}\ \mathscr{V}_b(2),$
      - $\mathrm{S}_1 \ \xrightarrow{\hbar}\ \mathrm{S}_2.$
- And
  - So
    - ▷ The:
      - Computation

to

- Generate
  - The i^{th}
    - ▷ Tuple

can

- Be
  - Written
    - ▷ As:
      - $(\mathscr{V}_a(i-1),\ \mathscr{V}_b(i-1),\ \mathrm{S}_{i-1}) \ \xrightarrow{\mathscr{V}_a}\ \mathscr{V}_a(i),$
      - $(\mathscr{V}_a(i-2),\ \mathscr{V}_b(i-2),\ \mathrm{S}_{i-1}) \ \xrightarrow{\mathscr{V}_a}\ \mathscr{V}_b(i),$
      - $\mathrm{S}_{i-1} \ \xrightarrow{\hbar}\ \mathrm{S}_i.$

Statement 7. *Any*

- *String*
  - *Can*

- ▷ *Be:*
  - *Given*

to:

"a Turing machine."

So assume that,

- We have
  - Given:
    - ▷ A string
      - To:

$$M.$$

- Then
  - Since:

$$M$$

- Always:

"*starts*"

from

- The
  - Left
    - ▷ Most:
      - Alphabet,
- And
  - From:
    - ▷ Statements 6 and 7
      - And Lemma 6,

we see that:

$$\mathrm{S}_0$$

will

- Be
  - *One*
    - Among:

$$\text{“}w \quad \textit{strings.}\text{”}$$

- And so
  - From:
    - Lemma 6,
      - And statement 5,

we see that:

$$(\mathscr{V}_a(1), \;\; \mathscr{V}_b(1))$$

will

- Be
  - *One*
    - Among:

$$\text{“}w \quad 2\text{-}\textit{tuples.}\text{”}$$

- And so
  - There
    - *Exists*:
      - A case

in

- Which

- The:

$$\text{“}\textit{first tuple}\text{”}$$

will

- Fail
    - A property,
        - Say $\mathscr{P}$
            - And *vice versa.*
- Then
    - From:
        - Statement 5,

we see that,

- When:

$$M$$

generates

- The
    - *Second*:
        - Tuple,

there

- Will
    - Be:
        - *No*
            - Computation

other

- Than

- That
    - Done
        - Using:

$$\mathscr{V}_a, \quad \mathscr{V}_b \quad \textit{and} \quad \hbar.$$

- And
    - So
        - When:

$$M$$

- Chooses:

$$\mathrm{S}_1,$$

from

- Statements 6 and 7,
    - And:
        - Lemma 6,

we see that:

$$\mathrm{S}_1$$

will

- Be
    - *One*
        - Among:

$$\text{“}w \quad \textit{strings.}\text{”}$$

- And
    - So
        - From:

- This,

- And
  - Since:

$$(\mathscr{V}_a(1),\ \ \mathscr{V}_b(1))$$

will

- Be
  - *One*
    - ▹ Among:

$$\text{“}w \quad 2\text{-}\textit{tuples},\text{”}$$

- And
  - From:
    - ▹ Lemma 6,
      - And statement 5,

we see that:

$$(\mathscr{V}_a(2),\ \ \mathscr{V}_b(2))$$

will

- Be
  - *One*
    - ▹ Among:

$$\text{“}w \quad 2\text{-}\textit{tuples}.\text{”}$$

- And so
  - There
    - ▹ *Exists*:
      - A case

in

- Which
  - The:

$$\text{“}\textit{second tuple}\text{”}$$

will

- Fail
  - The
    - ▹ Property: $\mathscr{P}$
      - And *vice versa.*
- And
  - So
    - ▹ In
      - General:

$$\mathrm{S}_{i-1}, \quad (\mathscr{V}_a(i-2),\ \mathscr{V}_b(i-2)) \quad \textit{and} \quad (\mathscr{V}_a(i-2),\ \mathscr{V}_b(i-2))$$

will

- Be
  - *One*
    - ▹ Among:
      - w things.
- And
  - So
    - ▹ In
      - General:

$$(\mathscr{V}_a(i),\ \mathscr{V}_b(i))$$

will

- Be
  - *One*
    - Among:

$$\text{“}w \quad 2\text{-}tuples.\text{”}$$

- And so
  - There
    - *Exists*:
      - A case

in

- Which
  - The:

$$\text{“}i^{th} \quad tuple\text{”}$$

will

- Fail
  - The
    - Property: $\mathscr{P}$
      - And *vice versa.*

Statement 8. *Let:*

$$M$$

- *Be:*

“a Turing machine”

- *And:*

$$\mathscr{P}$$

- *Some:*

"property."

- *Then*
  - *There*
    - *Will:*
      - *Exist*

a

- *Case*
  - *In:*
    - *Which,*
      - *The:*

"i^{th} element"

generated

- *By:*

$$M$$

will

- *Fail:*

$$\mathscr{P}.$$

- *And also*
  - *There*
    - *Will:*
      - *Exist*

a

- *Case*
  - *In:*
    - *Which,*
      - *The:*

$$\text{“}i^{th} \quad \textit{element”}$$

generated

- *By:*

$$M$$

will

- *Satisfy:*

$$\mathscr{P}.$$

3.2 Infinite machine combinatorics

Let:

$$M_i$$

- Be:

$$\text{“}\textit{a Turing machine.}\text{”}$$

- Then
  - We
    - Can
      - Change:

$$M_i$$

to

- Get:

$$M_{i+1}.$$

- And
  - So:
    - We

can

- Define
  - A sequence
    - Of:
      - *Machines.*
- In
  - This:
    - Sub section,

we

- Deal
  - With:
    - A Turing machine

that

- Generates
  - A sequence
    - Of:
      - *Machines.*
- In
  - Sub section 3.1,

we saw that,

- A program
  - Can
    - ▹ Be:
      - *Divided*

into

- A finite
  - Number
    - ▹ Of:
      - *Divisions*,

such that

- Each:
  - *Division*
    - ▹ Will:
      - *Correspond*

to

- A *variable*
  - Of
    - ▹ The:
      - *Machine*.
- And
  - So
    - ▹ From:
      - This,
- And

- Since
    - A machine:
        - Has

only:

- An initial state,
    - A program,
        - And some
            - Terminal states,

we see that,

- A machine
    - Can:
        - Only
            - Be:

“*modified*”

by:

- Removing
    - Variables
        - From:
            - *It*,
- Or
    - Changing
        - *Its*:
            - Variables,
- Or by
    - Adding

▷ New:

– Variables.

- And

  ∘ So

    ▷ Let:

$$\mathscr{V}_1^i, \quad \mathscr{V}_2^i, \quad \ldots, \quad \mathscr{V}_n^{i-1}, \quad \mathscr{V}_n^i$$

be

- The

  ∘ Variables

    ▷ In:

$$M_i.$$

- And

  ∘ Also

    ▷ Let:

$$\mathscr{V}(M_i) \quad = \quad \{ \quad \mathscr{V}_1^i, \quad \mathscr{V}_2^i, \quad \ldots, \quad \mathscr{V}_n^{i-1}, \quad \mathscr{V}_n^i, \quad \}.$$

- And

  ∘ Assume

    ▷ That:

$$M$$

had

- Generated:

$$M_i$$

- From:

$$M_{i-1}.$$

- And
  - So
    - ▷ From:
      - This,

we see that,

- With
  - Respect
    - ▷ To:

$$M,$$

the

- Ordinal:
  - Position
    - ▷ Of:

$$\mathscr{V}_1^i$$

- In:

$$\mathscr{V}(M_i),$$

will

- Always
  - Be: $1,$
    - ▷ And
      - Never: 3 *or* $5.$
- And
  - Similarly,

the

- Ordinal:
  - Position
    - Of:

$$\mathscr{V}_k^i$$

- In:

$$\mathscr{V}(M_i),$$

will

- Always
  - Be: k,
    - And
      - Never: $k+2$ *or* $k+8$.
- Then
  - Since:

$$M$$

- Generated:

$$M_i$$

- From:

$$M_{i-1},$$

we see that,

- There
  - Will
    - *Exit*:

- A function,

say:

$$\mathfrak{V}_a,$$

such that

- It
  - Will
    - Take
      - Some:

"*elements*"

- Of:

$$\mathscr{V}(M_i),$$

- And return
  - Another
    - Set
      - Of:

"*variables.*"

- Also
  - For
    - The sake:
      - Of:

"*simplicity,*"

assume

- That:

$$M$$

will

- Only
  - Use:

$$M_{i-1}$$

to

- Generate:

$$M_i.$$

- And
  - Also
    - That:

$$\mathfrak{V}_a$$

has

- Only
  - *Two*
    - Parameters:
      - In it.
- Then
  - From:
    - Statement 5,

we see that,

- Only:

$$\mathfrak{V}_a$$

will

- Be:
  - Used
    - To
      - Perform

the:

"computation"

which

- Modifies:

"a variable."

- And
  - Also
    - There:
      - Will

be

- Some
  - Precise:

"rules"

to

- Choose
  - The
    - Parameters
      - For:

$$\mathfrak{V}_a.$$

- And so

- In accordance
  - With:
    - Statement 4,

for

- *Two*
  - *Elements*
    - Of:

$$\mathscr{V}(M_{i-1}),$$

- Say:

$$\mathscr{V}_k^{i-1}, \ \ \mathscr{V}_{k+1}^{i-1}$$

- Let:

$$(\mathscr{V}_k^{i-1}, \ \ \mathscr{V}_{k+1}^{i-1}, \ \ \mathrm{S}_{i-1}) \ \ \xrightarrow{\mathfrak{V}_a} \ \ \mathscr{V}_k^i,$$

- Where:

$$\mathrm{S}_{i-1}$$

is

- The
  - $i-1^{th}$ string
    - Chosen
      - By

the

- Function:

$$\hbar$$

which

- We
  - Saw:
    - In
      - Sub section 3.1.

Or we see that,

- When
  - The i^{th} machine
    - Is:
      - Generated,

the

- Function:

$$\mathfrak{V}_a$$

will

- Take:
  - The k^{th}
    - And
      - The $k+1^{th}$

"*variables*"

- Of:

$$\mathscr{V}(M_{i-1})$$

- As:

"*parameters*,"

- And
  - Replace:

$$\mathscr{V}_k^{i-1}$$

- With:

$$\mathscr{V}_k^i,$$

so

- As
  - To
    - Get:

$$M_i.$$

- But
  - If:
    - There is
      - *No*:

$$\mathscr{V}_{k+1}^{i-1}$$

- Then:

"*a constant,*"

- Say:

$$\mathcal{C}_1$$

inbuilt

- Into:

$$M$$

will

- Be:

- Used
  - Instead
    - Of:

$$\mathscr{V}_{k+1}^{i-1}.$$

- But
  - For
    - The
      - Sake

of:

"*simplicity*,"

we

- Assume
  - That:

$$\mathscr{V}_{k+1}^{i-1}$$

is

- Always:

"*available*."

- Note
  - That,
    - Even
      - Though:

$$\mathscr{V}_{k}^{i-1} \qquad \textit{and} \qquad \mathscr{V}_{k+1}^{i-1}$$

- Maybe:

"*intertwined*,"

we see that,

- Some
  - Parts
    - Of:

$$\mathscr{V}_k^{i-1} \qquad \textit{and} \qquad \mathscr{V}_{k+1}^{i-1}$$

will

- *Not*
  - Be:

“*intertwined*,”

since

- If
  - *Not*,
    - Then:

$$\mathscr{V}_k^{i-1} \qquad \textit{and} \qquad \mathscr{V}_{k+1}^{i-1}$$

will

- Be
  - The:
    - Same.
- And
  - So
    - We
      - Can:

“*change*”

that

- Part
  - Of:

$$\mathscr{V}_k^{i-1}$$

which

- Is
  - *Not*:

“*intertwined*,”

- With:

$$\mathscr{V}_{k+1}^{i-1}$$

without

- Affecting
  - The:
    - Other.
- And
  - So
    - From
      - Statement 5,

in:

“*this case*”

we see that,

- The
  - Definition

- ▹ Of:

$$\mathfrak{V}_a$$

will

- Be
  - To
    - ▹ Make
      - Some:

"*changes*"

on

- That
  - Part
    - ▹ Of:

$$\mathscr{V}_k^{i-1}$$

which

- Is
  - *Not*
    - ▹ Intertwined
      - With:

$$\mathscr{V}_{k+1}^{i-1}.$$

- But
  - If
    - ▹ The:
      - Definition

of:

$$\mathfrak{V}_a$$

where

- To
  - Be:

$$\text{“}\textit{different}\text{”}$$

then

- There
  - Would:
    - ▹ Have

been

- Changes
  - In
    - ▹ Both:

$$\mathscr{V}_k^{i-1} \qquad \textit{and} \qquad \mathscr{V}_{k+1}^{i-1}.$$

- Or
  - If:
    - ▹ The
      - Definition

of:

$$\mathfrak{V}_a$$

where

- To
  - Be:

“*different*”

then

- An example
  - For
    - The:
      - Change

can

- Be
  - To
    - Swap:

$$\mathscr{V}_k^{i-1} \qquad \textit{and} \qquad \mathscr{V}_{k+1}^{i-1}.$$

- And
  - That:
    - Would

be

- Equivalent
  - To:
    - Changing
      - Both:

$$\mathscr{V}_k^{i-1} \qquad \textit{and} \qquad \mathscr{V}_{k+1}^{i-1}.$$

- And
  - So
    - The
      - Function:

$$\mathfrak{V}_a$$

will

- *Not*
  - Change
    - ▹ The ordinal:
      - Positions

of

- The
  - Elements
    - ▹ In:

$$\mathscr{V}(M_{i-1}).$$

- And
  - So
    - ▹ Let:

$$\mathfrak{V}_a(i)$$

be

- Used
  - To:
    - ▹ Denote

the

- i^{th} value
  - Returned
    - ▹ By:

$$\mathfrak{V}_a.$$

So we see that,

- There
    - Will
        - *Exist*:
            - A function,

say:

$$\pi,$$

- Such that:

$$\pi,$$

will

- Choose:
    - The
        - Parameters
            - For:

$$\mathfrak{V}_a.$$

- Then we see that,
    - The
        - Ordinal:
            - Position

of

- The
    - Variable
        - Chosen
            - By:

$\pi,$

may

- Always
    - Remain
        - ▹ The:
            - Same,
- Or
    - It
        - ▹ May:
            - Vary.
- But
    - If
        - ▹ It
            - Will *not*:

"*vary*,"

then

- Since
    - A variable
        - ▹ Is:
            - Made up

of

- A finite
    - Number
        - ▹ Of:

"*quintuples*,"

- And
    - Since
        - The:
            - Number

of

- Ways
    - To
        - Construct:
            - A quintuple

is:

“*a constant*,”

we see that,

- After
    - Sometime:

$$M$$

will

- Be
    - Forced
        - To:
            - Increase,

the

- Number
    - Of
        - Quintuples
            - In:

“*the variable.*”

- Then
    - Since
        - The
            - Number

of

- Quintuples
    - In:
        - A variable
            - Is *always*:

“*finite,*”

we see that,

- Even
    - Though
        - We:
            - *Increase*

the

- Number
    - Of
        - Quintuples:
            - In it,

we

- Cannot
    - Do
        - It:

        - Beyond

a

- Certain:

"limit."

- And
    - So
        - From:
            - This,
- And
    - Since
        - The:
            - Number

of

- Ways
    - To
        - Construct:
            - A quintuple

is:

"a constant,"

we see that,

- If
    - There
        - Is:

"no change,"

in

- The ordinal
    - Position
        - Of
            - Variables:

$$\text{“}\textit{chosen}\text{”}$$

- By:

$$\pi$$

then

- Only
    - A finite
        - Number
            - Of:

$$\text{“}\textit{changes}\text{”}$$

will

- Be
    - Made
        - In
            - The:

$$\text{“}\textit{entire process}\text{”}$$

of

- Generating
    - The
        - Infinite:

- Sequence,

which

- Inturn
  - Would
    - Mean:
      - That,

only

- A finite
  - Number
    - Of:
      - Machine

are:

"*generated.*"

- And so
  - The
    - Ordinal:
      - Position

of

- The
  - Variable
    - Chosen
      - By:

$$\pi$$

will

"*vary.*"

- And
    - So
        - Let:

$$\mathscr{V}_a^i \quad \textit{and} \quad \mathscr{V}_b^i,$$

be

- Two
    - Variables
        - Of:

$$M_i.$$

- Then
    - It
        - Maybe
            - That:

$$M$$

will

- Change:

$$\text{“}\mathscr{V}_a^i \quad \textit{twice,}\text{”}$$

- And
    - Then
        - Change:

$$\text{“}\mathscr{V}_b^i \quad \textit{twice}\text{”}$$

and so on.

- Or

- We
  - Can
    - Write:

$$\ldots \quad \rightarrow \quad \mathscr{V}_a^i \quad \rightarrow \quad \mathscr{V}_a^{i+1} \quad \rightarrow \quad \mathscr{V}_b^{i+2} \quad \rightarrow \quad \mathscr{V}_b^{i+3} \quad \rightarrow \quad \ldots$$

- And
  - So
    - We see that,
      - If:

$$\text{“}\mathscr{V}_a^i \quad \textit{satisfies}\text{”}$$

- Some:

$$\text{“}\textit{condition}\text{”}$$

- In:

$$\mathfrak{V}_a,$$

- Then:

$$\mathscr{V}_a^{i+1}$$

will

- Be:

$$\text{“}\textit{generated}\text{”}$$

- By:

$$\mathfrak{V}_a.$$

- And:

$$\mathscr{V}_a^{i+1}$$

will

- *Not*
    - Satisfy
        - Any:
            - Condition

in:

$$\mathfrak{V}_a.$$

- And
    - Similarly
        - For:

$$\mathscr{V}_b^{i+2} \qquad \textit{and} \qquad \mathscr{V}_b^{i+3}.$$

- And
    - Also
        - From:
            - Statement 4,

we see that,

- The
    - Number
        - Of
            - Those:

"*conditions*"

will

- Always be:
    - *Fixed*

        - ▹ And
            - *Finite.*

- And
    - So
        - ▹ From:
            - These,

we see that:

$$\mathfrak{V}_a(i-1),$$

should

- Be:

“*a parameter*”

- For:

$$\pi,$$

- And
    - Those
        - ▹ Conditions
            - In:

$$\mathfrak{V}_a$$

can

- And
    - Should
        - ▹ Be:
            - Copied

into:

$$\pi.$$

- And
  - In:

$$\pi,$$

- If
  - The
    - Parameter:

$$\text{“}\mathfrak{V}_a(i-1) \quad \textit{satisfies}\text{”}$$

- Those:

“*conditions*”

- And:

$$\mathrm{S}_{i-1}$$

also

- Agrees
  - With:
    - It,
- And
  - If:

$$\pi(i)$$

is

- Used
  - To:
    - Denote

the

- i^{th} value
  - Returned
    - By:

$$\pi,$$

- Then:

$$\pi(i) \quad = \quad \pi(i-1),$$

- Else:

$$\pi$$

will

- Use
  - Some
    - Finite
      - Criteria:

"*inbuilt in* π,"

to

- Choose
  - An
    - Element
      - Of:

$$\mathscr{V}(M_{i-1}).$$

- And
  - So

- ▹ The:
  - Computation

done

- Using:

$$\pi$$

in

- The i^{th}
  - Step
    - ▹ Can be:
      - Written

as:

$$(\mathfrak{V}_a(i-1),\ \mathscr{V}(M_{i-1}),\ \mathrm{S}_{i-1}) \xrightarrow{\pi} \mathscr{V}(M_{i-1}).$$

- And so
  - In
    - ▹ This:
      - Case,

we see that,

- If:

$$\pi$$

- Chooses:

$$\mathscr{V}_k^{i-1},$$

- Then:

$$\mathscr{V}_k^{i-1} \quad \textit{and} \quad \mathscr{V}_{k+1}^{i-1}$$

will

- Be:
  - The
    - ▹ Parameters
      - For:

$$\mathfrak{V}_a.$$

Then we see that,

- Since
  - A finite:
    - ▹ Number

of

- Quintuples
  - Are:
    - ▹ Used

to

- Represent:

“*a variable*”

- And
  - Since:
    - ▹ *Zero*
      - Number

of

- *Quintuples*
  - Will

- ▹ Never
  - – Represent:

"*a variable*,"

- And
  - ◦ Since
    - ▹ Only:
      - – A finite number

of

- *Alphabets*
  - ◦ Can
    - ▹ Be:
      - – *Written*

on

- The
  - ◦ Tape
    - ▹ Of:

$$M,$$

- And
  - ◦ Since
    - ▹ The:
      - – Number

of

- *States*
  - ◦ In:
    - ▹ A *machine*

- Is always:

"*a constant*"

we see that,

- If
  - We
    - Are:
      - *Not* going

to

- Increase
  - The:
    - *Cardinality*
      - Of:

"*the program*"

then

- We
  - Can
    - Only:
      - Make

a

- *Finite*
  - Number
    - Of:
      - *Changes*

in:

"*the program.*"

- And
    - So
        - ▹ We
            - Will:

"*run out,*"

- Of:

"*options*"

- After:

"*sometime.*"

- And
    - So
        - ▹ If we:
            - Keep

on

- Making
    - Changes
        - ▹ In:
            - The program,

then

- Initially,
    - Sequences:
        - ▹ Corresponding

to

- Variables

- Will:

"*change.*"

- And
    - Then
        - ▹ After:
            - Sometime,

we

- Cannot
    - Make
        - ▹ Anymore:
            - *Changes*

unless

- We:

"*increase*"

the

- Number
    - Of:

"*quintuples.*"

- And
    - So:
        - ▹ We

will

- Be:

- *Forced*
  - To:

“*increase*”

the

- Number
  - Of:
    - *Quintuples*
      - Or *variables.*
- And so:
  - When
    - The i^{th}
      - Machine

is:

“*generated,*”

we see that,

- If:

$$M$$

can

- And
  - Decides
    - To:
      - Change

“*the variables*”

then

- It
  - Will
    - Change
      - Some:

"*variables*,"

- And
  - Even
    - If:
      - It

can

- Make
  - Some
    - Changes
      - In:

"*the variables*,"

- But
  - Decides
    - *Not* to:
      - Do so,

then

- It will
  - Add
    - One
      - Or some:

"*variables*,"

- And
  - If:
    - ▹ It

cannot

- Make
  - Any
    - ▹ More:
      - Changes

in:

“*the variables*,”

then

- It
  - Will
    - ▹ Add:
      - One
      - Or some

“*variables*.”

- And
  - So:
    - ▹ There
      - Will *exist*:

“*a function*,”

- Say:

$\mathfrak{X}$,

such that

- At
  - Each:
    - ▹ Step,

it

- Will
  - Generate
    - ▹ A set
      - Of:

“*variables.*”

- And
  - So
    - ▹ If:

$$\mathfrak{X}(i)$$

is

- Used
  - To:
    - ▹ Denote

the

- i^{th} value
  - Returned
    - ▹ By:

$$\mathfrak{X},$$

then

- The
  - Computation
    - Done
      - Using:

$$\mathfrak{X}$$

in

- The i^{th}
  - Step
    - Can

be

- Written
  - As:

$$(\mathscr{V}(M_{i-1}),\ \ \pi(i-2),\ \ \mathrm{S}_{i-1}) \quad \xrightarrow{\mathfrak{X}} \quad \mathfrak{X}(i).$$

- The
  - Value
    - Of:

$$\mathfrak{X}(i)$$

will

- Also
  - Depend
    - On:

$$\pi(i-2)$$

since

- It
    - Need
        - And can:
            - Know

whether

- All
    - Possible
        - Modification
            - Have been:

$$\text{“}\textit{made}\text{”}$$

from

- The
    - Value
        - Of:

$$\pi(i-2).$$

- Then we see that,
    - Since
        - The:
            - Definition

of:

$$M$$

is

- To
    - Generate

- ▹ A sequence
  - Of:

"*machines*,"

- And
  - Since
    - ▹ These:
      - Functions,

are

- Enough
  - To
    - ▹ Do:
      - It,

we see that,

- There
  - Will
    - ▹ Be
      - *No* other:

"*function*"

other

- Than:
  - *These*
    - ▹ In:

$$M.$$

- And

- So
  - ▹ If:

$$\mathfrak{X}$$

has

- To:

$$\text{“\textit{know}”}$$

whether

- All
  - Possible
    - ▹ Modification
      - Have been:

$$\text{“\textit{made},”}$$

then

- It
  - Has
    - ▹ Use:
      - The values

of:

$$\pi(i-2) \qquad \textit{and} \qquad \mathfrak{V}_a(i-2).$$

- Then
  - Since:

$$\mathfrak{V}_a(i-2)$$

can

- Be
  - Deduced
    - From:

$$\pi(i-2)$$

- We see that:

$$\mathfrak{X}$$

only

- Need:

$$\pi(i-2).$$

- Then
  - Since
    - It
      - Is:

"*essential*"

- That:

$$\mathfrak{X}$$

know

- When
  - To
    - Add
      - New:

"*variables,*"

- And

- Since:

$$M$$

generates

- A sequence
  - Of:

"*machines*,"

- And
  - Since:

$$\pi,$$

is

- A finite:

"*something*,"

we see that,

- All:
  - *Rules*
    - In:

$$\pi,$$

can

- Be
  - Copied
    - Into:

$$\mathfrak{X},$$

- And
    - They:
        - Can

be

- Used
    - In:
        - Such
            - A way,

that:

$$\mathfrak{x}$$

can

- Know
    - Whether
        - All
            - Possible:

“*modification*”

have

- Been
    - Made
        - From:
            - The value

of:

$$\pi(i - 2).$$

- But

- If:
  - There is
    - *No*:

$$\pi(i-2)$$

then

- A constant,
  - Say:

$$\mathcal{C}_3$$

inbuilt

- Into:

$$M$$

will

- Be:
  - Used.
    - Instead
      - Of:

$$\pi(i-2).$$

- Also
  - We see that,
    - If:

$$\pi(i-2)$$

says

- That,

- Some
  - More:

$$\text{“}\textit{changes}\text{”}$$

can

- Be:

$$\text{“}\textit{made,}\text{”}$$

- But:

$$\mathfrak{X}$$

decides

- On:
  - The
    - Contrary,
- Then:

$$\mathfrak{X}(i) \quad = \quad \emptyset.$$

- And
  - So
    - If:

$$\mathfrak{X}(i) \quad = \quad \emptyset,$$

then

- There
  - Will
    - Be:
      - *No* increase

in

- The
    - Number
        - Of:
            - Variables,
- But
    - If:
        - *Not*,
            - Then:

$$M_i \quad = \quad \mathscr{V}(M_{i-1}) \cup \mathfrak{X}(i)$$

Also we see that,

- When
    - We
        - Add
            - A new:

"*variable*,"

we

- Will
    - Get
        - The next
            - In:

"*the sequence*,"

- And also
    - When
        - We

– Change:

"*a variable*,"

we

- Will
  - Get
    - ▹ The next
      - In:

"*the sequence*,"

- And so
  - Both:
    - ▹ Adding
      - A new variable,
    - ▹ And changing
      - A variable,

are:

"*equivalent*"

to

- Generating
  - The:
    - ▹ Next
      - In:

"*the sequence*."

- And
  - So
    - ▹ We:

– Assume

that,

- When
    - A new
        - ▹ Variable
            - – Is:

"*added*,"

then

- There
    - Will
        - ▹ Be:
            - – *No* change

in

- The:
    - Old
        - ▹ Variables.
- And so
    - When
        - ▹ There
            - – Is:

"*no change*"

in

- The:
    - Old

- ▹ Variables,

we

- Assume:

$$\pi(i-2) \quad = \quad \pi(i-1).$$

- And
  - So
    - ▹ Let:

$$\mathrm{S}_0$$

be

- The *first*
  - String
    - ▹ Chosen
      - By:

$$\hbar.$$

- But
  - Since
    - ▹ At:
      - This point,

there

- Is
  - No:

$$M_0,$$

- And

- Since:

$$\chi(M)$$

will

- *Never*:

$$\text{"}\textit{change}\text{"}$$

we see that,

- At
    - This:
        - ▹ Point,
            - The functions:

$$\pi \qquad \textit{and} \qquad \mathfrak{V}_a$$

cannot

- Be:

$$\text{"}\textit{used}\text{"}$$

- But
    - Only
        - ▹ The
            - Function:

$$\mathfrak{X}$$

will

- Be
    - Used
        - ▹ To:
            - Generate:

$$M_1.$$

- And
  - So
    - ▹ The:
      - Computation

done

- To
  - Generate:

$$M_1,$$

can

- Be
  - Written
    - ▹ As:
      - $\mathrm{S}_0 \quad \xrightarrow{\mathfrak{x}} \quad \mathfrak{X}(1),$
      - $\mathrm{S}_0 \quad \xrightarrow{\hbar} \quad \mathrm{S}_1.$
- And
  - So:

$$M_1 \quad = \quad \mathfrak{X}(1).$$

- Then
  - When
    - ▹ We
      - Generate:

$$\pi(2),$$

we see that,

- Since
    - The
        - Definition
            - Of:

$$\pi$$

- Is:

$$(\mathfrak{V}_a(i-1),\ \ \mathscr{V}(M_{i-1}),\ \ \mathrm{S}_{i-1}) \quad \xrightarrow{\pi} \quad \mathscr{V}(M_{i-1}),$$

- And
    - Since
        - At:
            - This point,

there

- Is
    - *No*:

$$\pi(1),$$

we see that,

- There
    - Will
        - Also be
            - *No*:

$$\mathfrak{V}_a(1).$$

- And
    - So
        - In

- The function:

$$\pi,$$

we

- Use
  - A constant
    - Say:

$$\mathcal{C}_3,$$

instead

- Of:

$$\pi(1),$$

- And:

$$\mathcal{C}_2,$$

instead

- Of:

$$\mathfrak{V}_a(1),$$

such that

- All
  - These
    - Constants:
      - Will

be

- Into:

$$M.$$

- And
  - So
    - ▹ The:
      - Computation

done

- To
  - Generate:

$$M_2$$

can

- Be
  - Written
    - ▹ As:
      - $(\mathscr{V}(M_1),\ \mathcal{C}_3,\ \mathrm{S}_1) \ \xrightarrow{\mathfrak{x}}\ \mathfrak{X}(2),$
- And if:
  - $\mathfrak{X}(2) \ = \ \emptyset,$
    - ▹ Then:
      - $(\mathcal{C}_2,\ \mathscr{V}(M_1),\ \mathrm{S}_1) \ \xrightarrow{\pi}\ \mathscr{V}_k^{M_1},$
      - $(\mathscr{V}_k^1,\ \mathscr{V}_{k+1}^1,\ \mathrm{S}_1) \ \xrightarrow{\mathfrak{V}_a}\ \mathscr{V}_k^2,$
      - $M_2 \ = \ (\mathscr{V}(M_1) \setminus \mathscr{V}_k^1) \ \cup \ \mathscr{V}_k^2,$
- But
  - If: *not*
    - ▹ Then:
      - $M_2 \ = \ \mathscr{V}(M_1) \ \cup \ \mathfrak{X}(2).$
- And:
  - Then:

▷ $\mathrm{S}_1 \quad \xrightarrow{\hbar} \quad \mathrm{S}_2$.

- And
  - So
    - ▷ In
      - General,

the

- Computation
  - Done
    - ▷ To:
      - Generate:

$$M_i$$

can

- Be
  - Written
    - ▷ As:
      - $(\mathscr{V}(M_{i-1}),\ \pi(i-2),\ \mathrm{S}_{i-1}) \quad \xrightarrow{\mathfrak{X}} \quad \mathfrak{X}(i)$,
- And if:
  - $\mathfrak{X}(i) \quad = \quad \emptyset$,
    - ▷ Then:
      - $(\mathfrak{V}_a(i-1),\ \mathscr{V}(M_{i-1}),\ \mathrm{S}_{i-1}) \quad \xrightarrow{\pi} \quad \mathscr{V}_k^{M_{i-1}}$,
      - $(\mathscr{V}_k^{i-1},\ \mathscr{V}_k^{i-1},\ \mathrm{S}_{i-1}) \quad \xrightarrow{\mathfrak{V}_a} \quad \mathscr{V}_k^{i}$,
      - $M_i \quad = \quad (\mathscr{V}(M_1) \setminus \mathscr{V}_k^{i-1}) \quad \cup \quad \mathscr{V}_k^{i}$,
- But
  - If: *not*
    - ▷ Then:
      - $M_i \quad = \quad \mathscr{V}(M_{i-1}) \quad \cup \quad \mathfrak{X}(i)$.

- And
    - Then:
        - ▷ $S_{i-1} \xrightarrow{\hbar} S_i$.
- Then
    - Since
        - ▷ There are
            - – *No* other:

$$\text{“\textit{functions}”}$$

other

- Than:
    - *These*
        - ▷ In:

$$M,$$

we see that,

- *No*
    - *Computation*
        - ▷ Other
            - – Than:

$$\text{“\textit{these}”}$$

will

- Be
    - Performed
        - ▷ In:
            - – Each step.

- Then
    - Using:
        - ▹ The same
            - Line

of:

"*arguments*"

which

- We
    - Gave:
        - ▹ In
            - Sub section 3.1,

we see that,

- When
    - There
        - ▹ Is
            - *No*:

"*change*"

in

- The
    - Number
        - ▹ Of:
            - Variables,

the

- i^{th} machine
    - Generated

- Will be
  - *One*

among:

"w_1 *machines,*"

- And
  - When
    - There
      - Is:

"*a change*"

in

- The
  - Number
    - Of:
      - Variables,

the

- i^{th} machine
  - Generated
    - Will be
      - *One*

among:

"w_2 *machines.*"

- And
  - Those:
    - *Two*
      - Integers:

$$w_1 \quad \textit{and} \quad w_2$$

will

- Always
  - Remain:
    - ▷ The
      - Same.
- But
  - For
    - ▷ The:
      - Sake

of:

"*simplicity,*"

we

- Assume
  - That:

$$w_1 = w_2 = w.$$

- And
  - So
    - ▷ The i^{th}
      - Machine:

"*generated*"

will

- Always
  - Be

- ▹ *One*
  - Among:

"w machines."

Statement 9. *Let:*

$$M$$

- *Be:*

"a Turing machine,"

such that

- *It:*

"generates"

a

- *Sequence*
  - *Of:*

"machines."

- *Then*
  - *At:*
    - ▹ *Anytime,*

the

- *Generated:*

"machine"

will

- *Always be:*

- *One*
  - *Among,*
    - *Say:*

$$\text{“}w \quad \textit{machines.”}$$

- *And*
  - *The:*
    - *Value*
      - *Of:*

$$w$$

will

- *Always*
  - *Remain:*

“a constant.”

4 Equivalence proof

Assumption 4. From

- Now
  - Onwards,
    - We:
      - Assume

that:

$$G_c$$

is

- The *one*

- And
  - *Only*:
    - Clique

of

- Size:

$$n//2$$

- In:

$$G.$$

Definition 10. S_G

- Is:
  - Defined

to

- Be
  - The
    - Set:

$$\{ \quad G_i \ : \quad G_i \subset G \quad \textit{and} \quad |V(G_i)| \quad = \quad n//2 \quad \}.$$

- And:

$$p \quad = \quad |S_G|.$$

Note that,

- The
  - Value
    - Of:

$$p$$

is

- Exponential
    - Compared
        - ▷ To:
            - The value

of:

$$n.$$

- And
    - So
        - ▷ If:
            - We

can

- Prove:
    - That,
        - ▷ There:
            - *Exists*

a

- Case
    - In
        - ▷ Which:

$$G_c$$

is

- The:
    - p^{th} subgraph:

- ▷ Chosen,
  - Then:

$$P \neq NP \quad \text{and} \quad NP = co\text{-}NP.$$

- In
  - This:
    - ▷ Section,

we

- Prove
  - That,
    - ▷ There:
      - *Exists*

a

- Case
  - In
    - ▷ Which:

$$G_c$$

is

- The
  - p^{th} subgraph:
    - ▷ Chosen,
- Thereby
  - Proving:

$$NP = co\text{-}NP.$$

- And

- So
  - In:
    - This section,

we

- Concentrate
  - Only:
    - On

the

- Part
  - Of:
    - MACHINE-STRUCTURE

that

- Encapsulates
  - The:
    - Heuristics.
- Also
  - In:
    - MACHINE-STRUCTURE,

we see that,

- The:

"*place*"

where

- The
  - Chosen

- Subgraph
  - Is:

"*checked*"

to

- Be:

"*a clique*"

of

- Size: $n//2$
  - Or
    - *Not*,
      - Resides

in:

$$\Phi.$$

- And
  - So
    - If:

$$\Phi$$

where

- To
  - Be:
    - A part

of

- The
  - Heuristics

    - ▹ In:
        - MACHINE-STRUCTURE,

then

- There
    - Will
        - ▹ Be:
            - Redundancies.
- And
    - So
        - ▹ We:
            - Can

assume

- That,
    - The:

“*heuristics*”

- In:

“MACHINE-STRUCTURE”

will

- *Not*
    - Take
        - ▹ Any:
            - Effort

to

- Check

- Whether
  - ▹ The chosen:
    - Subgraph

is:

"*a clique*"

of

- Size: $n//2$
  - Or
    - ▹ *Not*,
- But
  - It will
    - ▹ Simply:
      - Choose

"*a subgraph*,"

- And
  - Ask:

$$\Phi$$

to

- Check
  - Whether
    - ▹ The chosen:
      - Subgraph

is:

"*a clique*"

of

- Size: $n//2$
  - Or
    - *Not.*
- And
  - So
    - If we
      - Replace:

$$\Phi$$

by

- Another
  - Function,
    - Say:

$$\Omega,$$

- Then:

$$M$$

will

- Choose:

“*a subgraph*,”

- And
  - Check:
    - Whether

it

- Satisfies: Ω
    - Or
        - *Not.*
- And so
    - If
        - We
            - Replace:

$$\Phi \quad \textit{by} \quad \Omega,$$

then

- When:

$$M$$

performs

- The:

“*computation*,”

we

- Can
    - Note
        - Down:
            - All

the

- Subgraphs:
    - Chosen
        - In
            - The:

"*exact order,*"

- And
  - After:

$$M$$

has

- Finished:

"*the computation,*"

we

- Can:
  - Check
    - Whether:

$$G_c$$

was

- Chosen
  - In:
    - Polynomial
      - Time.
- And
  - So
    - We:
      - Assume

that,

- In:

“MACHINE-STRUCTURE,”

we

- Have
    - Replaced:

$$\Phi$$

by

- Another
    - Function,
        - Say:

$$\Omega,$$

such that

- *No*
    - Subgraph
        - Of:

$$G$$

will

- Satisfy:
    - It.
- The
    - Heuristics:
        - Used
            - In:

$$M$$

to

- Choose:

“*subgraphs*”

can

- Be:
    - Classified

into

- The finite,
    - And
        - Infinite
            - Cases.
- In
    - The:
        - Finite
            - Case

the

- Heuristic
    - Used
        - Is:
            - *Fixed*,
- And
    - Will *not*
        - Vary:
            - Over time.
- In

- The:
  - ▹ Infinite
    - – Case

a

- New:

“heuristics”

will

- Be
  - Generated
    - ▹ At:
      - – Each step.
- And
  - That
    - ▹ Generated:
      - – Heuristics

will

- Be
  - Used
    - ▹ To
      - – Choose:

“*the next subgraph.*”

- And so
  - In
    - ▹ The:
      - – Infinite case,

the

- Machine
    - Will
        - ▷ Modify:
            - Itself.
- The
    - Infinite
        - ▷ Case:
            - Can

be

- Further
    - Classified
        - ▷ Into:
            - $1^{st},\ 2^{nd},\ 3^{rd},\ \ldots\ r^{th}$ infinite cases,
            - The extreme infinite case,
            - And the extra ordinary infinite case.
- We
    - Will
        - ▷ Justify:
            - Why

these

- Are
    - The only
        - ▷ Possible:
            - Infinite cases.
- Also

- Since:
  - Lemma 5
    - Says

that,

- The
  - String
    - Given
      - To:

$$M$$

- Is:

“*a representation*”

- Of:

$$G,$$

we

- Freely
  - Say
    - That:

$$G$$

is

- Given
  - To:

$$M,$$

- And:

$$M$$

works

- On:

$$G.$$

Statement 10. *The*

- *Subgraph:*

$$G_c$$

can

- *Be:*
  - *Present*
    - *In*
      - *Any:*

“part”

- *Of:*

$$G.$$

4.1 Finite case

In

- This
  - Sub section,
    - We
      - Assume

that:

$$M$$

will

- *Not*
    - Modify:

$$\text{“}\textit{itself.}\text{”}$$

- And
    - So
        - ▷ The:
            - Heuristics

in:

$$M$$

will

- Always
    - Remain:
        - ▷ The
            - Same.
- And
    - So
        - ▷ When:

$$G$$

is

- Given
    - To:

$$M,$$

it

- Will
  - Initially
    - ▷ Choose:

$$\text{“}\textit{a subgraph}\text{”}$$

- Of:

$$G,$$

- And check
  - Whether:
    - ▷ It
      - – Satisfies:

$$\Omega.$$

- But
  - Since:
    - ▷ It

will

- *Not*
  - Satisfy:

$$\Omega,$$

the

- Machine:

$$M$$

will

- Choose

- Another
    - Subgraph.
- And
  - So
    - The:
      - Process

will:

"*continue.*"

- Then
  - Since:

$$M$$

- Uses:

"*a heuristic*"

to

- Choose:

"*subgraphs,*"

when

- We
  - Look
    - At
      - The:

"*computation*"

- In:

“*detail*,”

we see that,

- The
    - *First*
        - ▹ Subgraph:
            - Chosen

will

- Depend
    - On:
        - ▹ The part
            - Of:

$$G$$

from

- Which:

$$M$$

- Starts:

“*analyzing*,”

- And
    - The *second*
        - ▹ One:
            - Chosen

will

- Depend
    - On

- ▷ The value:
  - Returned

by

- The:

"*heuristics*,"

which

- Inturn
  - ○ May:
    - ▷ Depend

on

- The
  - ○ *First*
    - ▷ Subgraph.
- And
  - ○ So
    - ▷ The:

"*computation*"

done

- By:

$$M$$

will

- Be:

"*equivalent*"

to

- A particular:

“*sequence*”

- Of:

“*element*”

- Of:

$$S_G.$$

- And
  - So
    - The:

“*computation*”

done

- By:

$$M$$

will

- Be
  - Equivalent:
    - To

“*a sequence*”

- Of:

“*criteria*”

for

- Choosing
    - Subgraphs
        - Of:

$$G.$$

- In
    - Sub section 3.1,

we saw that,

- Computations:
    - Done:
        - On
            - The:

"*variables*"

- Of:

$$M$$

will

- Be used
    - To
        - Generate
            - The:

"*next.*"

- And so
    - In
        - This:
            - Case,

we see that,

- Computations
  - Done:
    - ▷ On
      - The:

"*variables*"

- Of:

$$M$$

will

- Be
  - Used
    - ▷ To:
      - Generate

a

- New:
  - Criteria
    - ▷ To
      - Choose:

"*the next subgraph.*"

But we see that,

- There
  - Maybe
    - ▷ Variables
      - In:

$$M$$

that

- Will
  - *Not*
    - Help
      - In:

$$\text{“}\textit{choosing}\text{”}$$

- An:

$$\text{“}\textit{element}\text{”}$$

- Of:

$$S_G.$$

- And so
  - For
    - The:
      - Time being,

assume:

Assumption 5. All

- Computations:
  - Done

using

- The
  - Variables
    - Of:

$$M$$

will

- *Only*
  - Give:
    - A criteria

for

- Choosing
  - An
    - Element
      - Of:

$$S_G.$$

We

- Will soon
  - Talk
    - About:
      - Cases

in

- Which
  - Some:
    - Variables

will

- *Not*:

“*help*”

in

- Choosing:

“*a subgraph*.”

- Then
    - From:
        - Statements 8 and 10,
            - And assumption 4,

we see that,

- There
    - Will
        - *Exist*:
            - A case,

in

- Which
    - The
        - Criteria
            - To:

“*choose*”

the

- *First*
    - Subgraph
        - Will:
            - Be

such that,

- It

- Will:
  - ▹ *Not*
    - Satisfy:

$$\Phi.$$

- And
  - Also:
    - ▹ In
      - General,

if:

$$M$$

has

- Chosen:

$$\text{“}g \quad \textit{subgraphs}\text{”}$$

- Where:

$$g \quad < \quad p-1,$$

then

- From:
  - Statements 8 and 10,
    - ▹ And assumption 4,

we see that,

- There
  - Will
    - ▹ *Exist*:
      - A case,

in

- Which
  - The
    - ▹ Criteria
      - To:

$$\text{“}\textit{choose}\text{”}$$

the

- *Next*
  - Subgraph
    - ▹ Will:
      - Be

such that,

- It
  - Will
    - ▹ *Not*
      - Satisfy:

$$\Phi.$$

- And so
  - In
    - ▹ This:
      - Case,

under

- The:
  - Assumption 5,

we see that,

- There
  - Will
    - *Exists*:
      - A situation

in

- Which:

$$G_c$$

is

- The:
  - p^{th} one
    - Chosen.
- But
  - It
    - Maybe:
      - The case

that,

- There
  - Are
    - Variables
      - In:

$$M$$

that

- Will

- *Not*:
    - Help

in

- Choosing
    - An
        - Element
            - Of:

$$S_G.$$

- But
    - Since
        - They
            - Are:

"*variables*,"

we see that,

- Some:

"*computations*"

will

- Be
    - Done
        - Using:
            - Them.

- In
    - Section 2,

we saw that,

- All
  - Computations:
    - Should

be

- Either:
  - To
    - Choose
      - A subgraph

of:

$$G,$$

- Or
  - Applying:

$$\Phi \qquad \textit{or} \qquad \Omega.$$

- And
  - So
    - If
      - Some:

"*variables*"

will

- *Not*
  - Help
    - In:
      - Choosing

an

- Element
    - Of:

$$S_G,$$

then

- Computations
    - Done
        - Using:
            - Them,

must

- Be
    - To:
        - *Repeat*

the

- Computations
    - Done:
        - Using
            - The other:

"*variables*."

But we see that,

- When
    - The
        - Computations:
            - For

the

- Variables
  - To
    - ▹ Choose:
      - A subgraph

is:

“*performed*,”

we

- Will
  - Get:

“*a criterion*”

to

- Choose:

“*a subgraph*.”

- Also we see that,
  - This
    - ▹ Repetition:
      - Has

to:

“*stop*,”

- Since:

“*a subgraph*”

has

- To

- Be:

“*chosen.*”

- And
    - So
        - If
            - We:

“*repeat*”

it

- A *fixed*
    - And
        - *Finite*
            - Number

of:

“*times,*”

then

- We
    - Will
        - Get
            - Another:

“*criterion*”

to

- Choose
    - Another:

“*subgraph.*”

- And
    - So
        - ▹ The:
            - Case,

when

- There
    - Is
        - ▹ A fixed:
            - Number

of:

"repetitions,"

will

- Be equivalent
    - To
        - ▹ The:
            - Case

where

- There
    - Are:
        - ▹ *No*
            - Repetitions.
- Also
    - If
        - ▹ There
            - Is:

“a change”

in

- The
  - Number
    - Of:
      - *Repetitions*,

then

- There
  - Will
    - Be:
      - Some

“rules”

for

- That:

“change.”

- And
  - Also
    - We
      - Will:

“get”

- Another:

“criterion”

in

- The

    - Sequence
        - Of:

“*criteria*”

that

- Would
    - Have
        - Been:
            - Generated

if

- There
    - Where:
        - *No*
            - Repetitions.
- And
    - So
        - The
            - Same:

“*arguments*”

which

- We
    - Gave:
        - Earlier,
            - Will

be:

“*applicable.*”

- And
  - So
    - The:
      - Case,

when

- There
  - Is:
    - A change

in

- The
  - Number
    - Of:
      - *Repetitions*,

will

- Be
  - Equivalent to:
    - The
      - Case

where

- There
  - Are:
    - *No*
      - Repetitions.
- And so
  - In
    - This
      - Case:

$$NP = co\text{-}NP \quad \textit{and also} \quad P \neq NP.$$

4.2 1^{st} infinite case

In

- This
  - Sub section,
    - ▹ We
      - Assume

that:

$$M$$

has

- An
  - Heuristic
    - ▹ Generator.
- And
  - So
    - ▹ When:
      - A graph

is

- Given
  - To:

$$M,$$

we see that,

- It
  - Will
    - ▹ Generate:
      - A heuristic,

- And
    - Use that
        - Newly
            - Generated:

"*heuristic*"

to

- Choose:

"*a subgraph.*"

- And
    - Then:
        - It

will

- Check:
    - Whether

the

- Chosen
    - Subgraph
        - Satisfies:

$$\Omega.$$

- But
    - Since:
        - It

will

- *Not*
    - Satisfy:

$$\Omega,$$

the

- Machine:

$$M$$

will

- Generate
    - Another:

"*heuristic*,"

- And
    - Choose
        - Another:
            - Subgraph,
- And
    - Check:
        - Whether

the

- Newly
    - Chosen
        - Subgraph
            - Satisfies:

$$\Omega.$$

- And
    - The
        - ▹ Process
            - Will:

"*continue.*"

- And
    - So
        - ▹ If: M
            - Runs

for

- An infinite
    - Amount
        - ▹ Of:
            - Time,

then

- It
    - Will:
        - ▹ Generate

the

- Infinite
    - Sequence
        - ▹ Of:
            - *Heuristics*:

$$\mathscr{H}_0, \quad \mathscr{H}_1, \quad \ldots, \quad \mathscr{H}_j, \quad \ldots. \tag{4}$$

- In

  - Sub section 3.1,

we saw that,

- The
  - Next
    - ▷ In:
      - A sequence

can

- *Only*
  - Be:
    - ▷ Generated

by

- Changing
  - The:
    - ▷ Previous.
- And so
  - For
    - ▷ The:
      - Time being,

assume:

Assumption 6. The

- Computations:
  - Done

using

- The

- Variables
  - Of:
    - Generator

will

- *Only*:
  - Help

in

- Modifying
  - The already
    - Existing:
      - Heuristics.

We

- Will soon
  - Talk
    - About:
      - Cases

in

- Which
  - Some:
    - Variables

will

- *Not*
  - Help
    - In
      - Generating:

“*heuristics.*”

- Also
  - If:

$$\mathscr{H}_i$$

is

- The i^{th}
  - Heuristics:
    - ▹ Generated,

then

- Assume
  - That:

$$w(i)$$

is

- The
  - Number
    - ▹ Of
      - Criteria:

$$\mathscr{H}_i$$

can

- Generate
  - At:
    - ▹ A time.
- Also

- We see that,
  - If:

$$w(i)$$

- Is:

"*a constant,*"

then

- There
  - Will
    - Be:
      - Only

a

- Finite:
  - Number
    - Of
      - Different

"*heuristics*"

in

- The:
  - Sequence 4.
- And
  - It
    - Will
      - Be:

"*equivalent*"

- To:

"*the finite case.*"

- Also
  - If:

"$w(i)$ *decreases,*"

then

- After sometime,
  - It:
    - Will
      - Be:

"*zero.*"

- And
  - So:

$$w(i)$$

will

- Only:

"*increase.*"

- And
  - Also
    - From:
      - Statement 9,

we see that:

$$\mathscr{H}_i$$

will

- Be
    - *One*
        - Among:

“w *heuristics*,”

- And
    - *Never*
        - A random:
            - *One*

from

- An
    - Infinite:
        - Set.
- And
    - Also
        - From:
            - Statement 8,

we see that,

- The:
    - Values
        - In
            - The

“*variables*”

- Of:

$$\mathscr{H}_i$$

will

- Be
    - *One*
        - Among:

$$\text{“}w(i) \quad \textit{values,}\text{”}$$

- And
    - From:
        - Statement 10,
            - And assumption 4,

we see that,

- Those:
    - Values

may

- Or
    - May
        - *Not*:
            - Give

a

- Criteria
    - To
        - Choose:

$$G_c.$$

- And

- So:

$$M$$

- Cannot:

"*generate*"

an

- Arbitrary
  - Random:
    - Heuristics,

that

- Will
  - Certainly
    - Choose:

$$G_c$$

in

- The
  - Next:

"*step.*"

- And
  - So
    - At:
      - Anytime,

the

- Chosen:

“*subgraph*”

will

- Always
  - Be
    - ▷ *One*
      - Among:

$$\text{“}w \times w(i) \quad \textit{subgraphs.}\text{”}$$

- And
  - So:
    - ▷ In
      - General,

if:

$$M$$

has

- Chosen:

$$\text{“}g \quad \textit{subgraphs}\text{”}$$

- Where:

$$g \quad < \quad p-1,$$

then

- There
  - Will:
    - ▷ *Exist*
      - A case,

in

- Which
    - The
        - Decision:
            - Made

to

- Choose
    - The:
        - Next

will

- Lead
    - To
        - Choosing:
            - A subgraph

which

- Will
    - *Not*
        - Satisfy:

$$\Phi.$$

- An so
    - In
        - This:
            - Infinite case,

under

- The:
    - Assumption 6,

we see that,

- There
  - Will
    - ▹ *Exists*:
      - A situation

in

- Which:

$$G_c$$

is

- The:
  - p^{th} one
    - ▹ Chosen.
- Then
  - Since
    - ▹ The:
      - Generator

will

- *Not*
  - Vary
    - ▹ Over:
      - Time,

we see that,

- If
  - The
    - ▹ Generator:

- Helps

in

- Choosing:

"*a subgraph*,"

then

- It
  - Will
    - Always:
      - Be

a

- Constant

"*help*."

- And so
  - If
    - The
      - Generator:

"*helps*"

in

- Choosing:

"*a subgraph*,"

then

- It
  - Will

        - Be:
            - Like,

a

- Constant
    - Thing
        - Is:
            - Always

a

- Part of:
    - The
        - Generated:
            - Heuristics.
- And so
    - We
        - Can:
            - Assume

that,

- The generator
    - Will
        - *Not*:
            - Help

in

- Choosing:

“*a subgraph*,”

- We

- Had:
  - ▹ Assumed

that,

- All
  - Computations
    - ▹ Will
      - Only:

“*help*”

in

- Modifying
  - The
    - ▹ Already
      - Existing:

“*heuristics.*”

- But
  - It
    - ▹ Maybe:
      - The case

that,

- There
  - Are
    - ▹ Variables
      - In:

“*the generator*”

that

- Will
    - *Not*:
        - Help

in

- Modifying:
    - The
        - Already
            - *Existing*:

“*heuristics*.”

- But
    - Since
        - They
            - Are:

“*variables*,”

we see that,

- Some:

“*computations*”

will

- Be
    - Done
        - Using:
            - Them.
- But we see that,

- ◦ All
  - ▹ Those:
    - – Variables

in:

"the generator"

which

- Will
  - ◦ *Not*:
    - ▹ Help

in

- Modifying
  - ◦ The
    - ▹ Present:
      - – Heuristics

should

- In
  - ◦ Some
    - ▹ Way:
      - – Help

in

- Generating
  - ◦ The
    - ▹ Next:
      - – Heuristics,

since

- The
  - Combined:

"*result*"

of

- The
  - Generator
    - Is
      - To:

"*generate*"

a

- New:

"*heuristic.*"

- And
  - So
    - If
      - Some:

"*variables*"

will

- *Not*
  - Help
    - In:
      - Modifying

the

- Already

- ◦ Existing:

"*heuristics*,"

then

- Computations
  - Done
    - ▹ Using:
      - Them,

must

- Be
  - To:
    - ▹ *Repeat*

the

- Computations
  - Done:
    - ▹ Using
      - The other:

"*variables*."

But we see that,

- When
  - The
    - ▹ Computations:
      - For

the

- Variables

- To
  - Generate:
    - A heuristic

is:

"performed,"

we

- Will
  - Get:

"a criterion"

to

- Generate:

"a heuristic."

- Also we see that,
  - This
    - Repetition:
      - Has

to:

"stop,"

- Since:

"a subgraph"

has

- To
  - Be:

"*chosen.*"

- And
    - So
        - If
            - We:

"*repeat*"

it

- A *fixed*
    - And
        - *Finite*
            - Number

of:

"*times,*"

then

- We
    - Will
        - Get
            - Another:

"*criterion*"

to

- Generate
    - Another:

"*heuristic.*"

- And

- So
  - ▹ The:
    - – Case,

when

- There
  - Is
    - ▹ A fixed:
      - – Number

of:

"*repetitions*,"

will

- Be equivalent
  - To
    - ▹ The:
      - – Case

where

- There
  - Are:
    - ▹ *No*
      - – Repetitions.
- Also
  - If
    - ▹ There
      - – Is:

"*a change*"

in

- The
  - Number
    - ▹ Of:
      - *Repetitions*,

then

- There
  - Will
    - ▹ Be:
      - Some

"*rules*"

for

- That:

"*change*."

- And
  - Also
    - ▹ We
      - Will:

"*get*"

- Another:

"*heuristics*"

in

- The
  - Sequence

▹ Of:

"*heuristics*"

that

- Would
  - Have
    - ▹ Been:
      - Generated

if

- There
  - Where:
    - ▹ *No*
      - Repetitions.
- And
  - So
    - ▹ The
      - Same:

"*arguments*"

which

- We
  - Gave:
    - ▹ Earlier,
      - Will

be:

"*applicable.*"

- And
    - So
        - The:
            - Case,

when

- There
    - Is:
        - A change

in

- The
    - Number
        - Of:
            - *Repetitions*,

will

- Be
    - Equivalent to:
        - The
            - Case

where

- There
    - Are:
        - *No*
            - Repetitions.
- And so
    - In
        - This
            - Case:

$$NP = co\text{-}NP \quad \textit{and also} \quad P \neq NP.$$

4.3 2^{nd}, 3^{rd}, $\ldots$, r^{th} infinite cases

In

- The 2^{nd}
    - Infinite
        - ▷ Case,

the

- Machine:

$$M$$

- Has:

"*an heuristic generator generator.*"

- Then
    - Similar:
        - ▷ To

the

- Previous
    - Infinite
        - ▷ Case,

it

- Can
    - Be
        - ▷ Shown:
            - That,

in

- The 2^{nd}

- Infinite
    - Case
        - Also:

$$NP \quad = \quad co\text{-}NP \qquad \textit{and also} \qquad P \quad \neq \quad NP.$$

- In
    - The 3^{rd}
        - Infinite
            - Case,

the

- Machine:

$$M$$

- Has:

“*an heuristic generator generator generator.*”

- And
    - Also
        - Similarly,

it

- Can
    - Be
        - Shown
            - That,

in

- The 3^{rd}
    - Infinite

- ▷ Case:

$$NP \quad = \quad co\text{-}NP \qquad \textit{and also} \qquad P \quad \neq \quad NP.$$

- In
  - The r^{th}
    - ▷ Infinite
      - Case,

the

- Machine:

$$M$$

- Has:

$$\text{“}\textit{an heuristic } \underbrace{\textit{generator} \ldots \textit{generator}}_{r \ \textit{generators}}\text{.”}$$

- And
  - Also
    - ▷ Similarly,

it

- Can
  - Be
    - ▷ Shown
      - That,

in

- The r^{th}
  - Infinite
    - ▷ Case:

$$NP \quad = \quad co\text{-}NP \qquad \textit{and also} \qquad P \quad \neq \quad NP.$$

4.4 Extreme infinite case

In

- This
    - Case,

the

- Machine:

$$M$$

starts

- From
    - One
        - Of:
            - The

"r^{th} *infinite case*,"

- And
    - Then
        - Moves:
            - On

to

- Some
    - Other
        - Infinite case,
            - Say:

"$r+8^{th}$ *infinite case*,"

and so on.

- But
  - Since:

$$M$$

has

- To
  - Stop
    - Generating
      - Heuristics

before

- It
  - Chooses
    - A subgraph,
- And
  - Since
    - The:
      - Computation

done

- By:

$$M$$

is

- A step
  - By
    - Step:
      - Process,

we see that:

$$M$$

will

- *Not*
    - Be
        - In
            - The:

$$\text{“}z + 1^{th} \quad \textit{infinite case,}\text{”}$$

for

- Some
    - Arbitrary:
        - Value
            - Of:

$$z,$$

during

- The
    - Entire:
        - Computational
            - Process.
- And
    - So
        - From:
            - This,
- And
    - Since:

$$\chi(M)$$

will

- Always
  - Be
    - The
      - Same

in:

"the beginning,"

- And
  - Since:

$$M$$

- Is:

"a deterministic something,"

- And
  - From:
    - What:
      - We

saw

- In
  - Sub sections 4.1 and 4.2,

we see that,

- In
  - All

▷ This

- Case:

$$NP \quad = \quad co\text{-}NP \qquad \text{and also} \qquad P \quad \neq \quad NP$$

- Also
  - This
    - ▷ Case
      - Can

be

- Extended
  - Into:
    - ▷ 1^{st} Extreme infinite case,
    - ▷ 2^{nd} Extreme infinite case,

      $\vdots$
- And
  - Also
    - ▷ The extreme extreme infinite case,
      - And 1^{st} extreme extreme infinite case,

        $\vdots$
- And so on.
- And
  - Also
    - ▷ Similarly,

in

- All
  - These

▹ Extended

– Cases,

we see that:

$$NP \quad = \quad co\text{-}NP \qquad \textit{and also} \qquad P \quad \neq \quad NP.$$

4.5 Extra ordinary infinite case

Until

- Now
  - We saw that,
    - ▹ Either:

the

- Machine
  - Will *not*
    - ▹ Modify:
      - – Itself,
- Or
  - It
    - ▹ Will:
      - – Modify

some

- Parts
  - Of:
    - ▹ It,
- And
  - The rest

      ▷ Will
        – Remain:

$$\text{“}\textit{unmodified.}\text{”}$$

- In
    ○ This case,
      ▷ Nothing
        – Of:

$$M$$

will

- Remain:

$$\text{“}\textit{unmodified.}\text{”}$$

- But
    ○ When
      ▷ It is:
        – So,

we see that,

- Only:

$$M$$

can

- And
    ○ Will
      ▷ Modify:
        – Itself,

- But

- When:

$$M$$

uses

- A quintuple
  - To
    - ▹ Perform:

"a computation,"

we see that,

- That
  - Quintuple
    - ▹ Will:
      - *Not*

"change"

until

- That:
  - Move
    - ▹ Has
      - Been:

"completed."

- And
  - So
    - ▹ If:

"nothing"

- Of:

$$M$$

will

- *Remain*:

"*unmodified*,"

- Then:

$$M$$

will

- Have
  - To
    - ▹ First:
      - *Construct*

a

- A new
  - Machine,
    - ▹ Say:

$$M',$$

- And
  - Then:

$$M'$$

will

- Modify:

$$M$$

to

- Get:

$$M'',$$

- And
  - Later:

$$M''$$

may

- Or
  - May
    - ▷ *Not*
      - – Modify:

$$M'$$

and so on.

- But
  - In
    - ▷ Sub section 3.2,

we saw that,

- All
  - These:

"*modifications*"

can

- Only
  - Be

    - ▷ Done
      - In:

"*a particular way.*"

- And
  - So
    - ▷ From:
      - This,

- And
  - Since:

$$M$$

- Is:

"*a deterministic something,*"

- And
  - Since:

$$\chi(M)$$

will

- Always
  - Be
    - ▷ The
      - Same

in:

"*the beginning,*"

- And

- Since:

$$M$$

will

- Have
  - To
    - Stop
      - All:

"*modifications*"

before

- Choosing:

"*a subgraph,*"

- And
  - Since
    - The:
      - Computation

done

- By:

$$M$$

is

- A step
  - By
    - Step:
      - Process,

we see that,

- When
    - A subgraph
        - ▹ Is:
            - Chosen,

$$M$$

will

- Always
    - Be
        - ▹ In
            - Some:

$$\text{“}r^{th} \quad \textit{infinite case.”}$$

- And
    - So
        - ▹ From:
            - What

we

- Saw
    - In
        - ▹ Sub sections 4.1 and 4.2,

we see that,

- In
    - This
        - ▹ Case
            - Also:

$$NP \quad = \quad co\text{-}NP \qquad \textit{and also} \qquad P \quad \neq \quad NP.$$

4.6 Proving the equivalence

We saw that,

- The
    - Computation
        - ▹ Done
            - By:

$$M$$

is

- A step
    - By
        - ▹ Step:
            - Process,
- And
    - So
        - ▹ From:
            - This,
- And
    - Since:

$$M$$

has

- To
    - Have:
        - ▹ An heuristics
            - In it,

we see that,

- The
  - Only
    - ▹ Possible
      - Cases,

when:

$$M$$

modifies

- Only
  - A part
    - ▹ Of:
      - *It*

are:

$$\textit{the} \quad 1^{st}, \quad 2^{nd}, \quad \ldots, \quad r^{th} \quad \textit{infinite cases}$$

$$\textit{and} \quad \textit{the extreme infinite cases.}$$

Then we see that,

- Either:

$$M$$

will

- *Not*
  - Modify:
    - ▹ *Itself,*
- Or modifies
  - A part
    - ▹ Of:

      - *It*,

- Or modifies
    - Everything
        - In:
            - *It*.

- But in
    - All
        - These:
            - Cases,

we saw that:

$$NP \quad = \quad co\text{-}NP \qquad \textit{and also} \qquad P \quad \neq \quad NP.$$

- And
    - So
        - We see that:

$$NP \quad = \quad co\text{-}NP \qquad \textit{and} \qquad P \quad \neq \quad NP.$$

- And
    - So
        - From:
            - This,

we see that:

$$P \quad \neq \quad co\text{-}NP.$$

References

[1] Hopcroft, John E.; Ullman, Jeffrey D., *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.